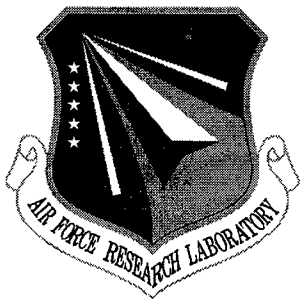
AFRL-IF-RS-TR-1998-192
Final Technical Report
September 1998



SECURE ENCRYPTION AND HIDING OF INTELLIGENCE DATA

Mission Research Corporation

Jiri Fridrich

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-1998-192 has been reviewed and is approved for publication.

APPROVED: Richard J. Simard

RICHARD J. SIMARD
Project Engineer

FOR THE DIRECTOR:

JOSEPH CAMERA, Deputy Chief
Information & Intelligence Exploitation Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFEC, 32 Brooks Rd, Rome, NY 13441-4114. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Sep 98	Final Apr 97 - Feb 98

4. TITLE AND SUBTITLE

SECURE ENCRYPTION AND HIDING OF INTELLIGENCE DATA

5. FUNDING NUMBERS

C - F30602-97-C-0209
PE - 65502F
PR - 3005
TA - 71
WU - 20

6. AUTHOR(S)

Jiri Fridrich

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Mission Research Corporation
1720 Randolph Road SE
Albuquerque, NM 87106-4245

8. PERFORMING ORGANIZATION REPORT NUMBER

MRC/ABQ-R-1856

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFEC
32 Brooks Rd
Rome, NY 13441-4114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-1998-192

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Richard J. Simard, IFEC, (315) 330-1798

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A method for hiding messages in the noise component of digital images in such a way that specific statistical characteristics of the carrier image are not disturbed is presented. The bits are injected in the secret message, taking into account the properties of the carrier image. Adaptive message hiding should be based on local properties of the carrier image. Secure steganographic techniques using digital images require message bits to be spread pseudo-randomly over the carrier image independently of the message lengths and not concentrated in the first rows or columns of the image. For this purpose, a simple but effective message-spreading algorithm based on parameterized chaotic maps was developed. The algorithm uses discretized two-dimensional chaotic maps as universal dispersers for random message scattering within the carrier image. This reports describes several new designs for secure watermarking schemes for digital images. The techniques offer a very high degree of robustness and security with respect to image modifications including noise and lossy compression. A technique is presented for overlaying the carrier image with a key-dependent image whose power is concentrated in the low frequencies. Message extraction is based on a discrete cosine transform. A second technique inserts bits into projections of image blocks onto random, smooth, orthogonal patterns individually generated for each image and user ID. This method avoids using publicly known basis functions, thereby increasing security against malicious attacks. Robustness test results against image distortion and manipulation techniques and security of attacks to removing hidden messages are presented. Patent application serial number PASN 08/986695, 8 Dec 97, Title "Digital Watermark" has been filed for this work.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Steganography, Data Embedding, Information Hiding, Digital Watermarking, Encryption, Information Warfare, Secure Communication, Copyright Protection, Authentication	64
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Executive Summary

The research, documented in this report, was carried out as part of a Phase I effort for the Department of Defense under the Small Business Innovative Research program. The objective of this research was to investigate the feasibility of developing techniques for secure and robust information hiding in digital imagery. Availability of efficient and secure algorithms for hiding secret, additional information in digital images can increase the communication bandwidth, contribute to an increased security of communication, and provide a unique and versatile protection of intellectual property rights, copyright authentication, and enable secure data dissemination.

Phase I research was conducted as four separate tasks. Task 1 consisted of developing steganographic methods in which the pixel values of a carrier image are slightly modified in order to encode a secret message. The pixel modifications need to be consistent with a noise model for that particular carrier image. This consistency is absolutely crucial in order to guarantee security of the hiding scheme. A general method was developed for noise models with symmetric probability distributions. It was also demonstrated how the security of data hiding schemes could be increased using chaotic maps as universal dispersers.

In Task 2, an extensive study of robust data embedding was performed. Several new robust data embedding techniques were developed, implemented in Matlab, and tested. The first robust data embedding technique is based on overlaying a pattern with its energy concentrated in low frequencies. The pattern is sensitive to every bit of the image and the author's ID. It is generated with a cryptographically strong pseudo-random number generator, processed with a cellular automaton, and smoothened with a smoothing kernel. The second method is based on modifying the values of discrete cosine coefficients, and the method does not require the original image for message extraction.

The third method is based on smooth, orthogonal patterns. It provides excellent robustness properties with a very high degree of security. Task 3 involved a study of security properties of the newly developed schemes. A new type of attack on robust data embedding schemes was discovered. The attack can be mounted against any non-adaptive watermarking scheme that embeds information into a selected subset of lowest frequency Fourier modes. It was shown that the attack could successfully break a simplified version of one of the most robust data embedding scheme described in [7]. The third robust data hiding scheme based on smooth orthogonal patterns was designed with the purpose to thwart this attack. The proposed data embedding schemes were tested against two general-purpose watermark-removing software packages: StirMark and UnZign [28]. UnZign did not pose any serious threat to the new schemes, while StirMark has shown that a special preprocessing of data prior to data detection is necessary. The security tests involved tests against collusion attack and overwatermarking.

The final Task 4 involved an extensive testing of the proposed data embedding schemes against intentional removal and against common image processing operations, such as resampling, blurring, noise adding, cropping, and lossy compression.

Phase I research was successful in demonstrating the feasibility and current capabilities of data hiding schemes. We were able to design non-robust steganographic schemes that preserve noise models of the carrier image, provided the probability distribution of the noise is symmetrical. In addition to that, we were able to develop robust data embedding schemes

with capabilities sufficient for unambiguous proving of ownership of digital images and fingerprinting multiple copies of an image (source-based watermarking). We have identified a limitation of robust data hiding schemes that do not require the original image for message extraction. Nonlinear geometric distortions, such as those introduced by StirMark, cause serious synchronization problems to such schemes. This problem could be overcome by using localized schemes with small blocks that would enable synchronization of the message detector. To confirm this observation and develop a robust scheme, further research is necessary.

Table of Contents

Table of Figures

List of Acronyms/Abbreviations/Quantities

DCT	Discrete Cosine Transformation
LSB	Least Significant Bit
PRNG	Pseudo-Random Number Generator
MRC	Mission Research Corporation
AFRL	Air Force Research Laboratory

Greek Symbols

α	Watermark Strength
η	Noise

1. Outline of Proposed Research

Before presenting in detail the results of the Phase I research, we outline the background and motivation for this research as indicated in the original Phase I SBIR proposal.

1.1 Background and Identification of Problem

The main goal of the research performed in Phase I was to prove the feasibility of new concepts for secure hiding of messages in digital images. The current research and the research proposed for the Phase II addresses Air Force deficiencies including, but not restricted to the following:

- MOB-XXX-96-INF-023, "Intelligence Collection and Imagery Dissemination";
- ISR-IFAP-96-XX-042, "Lack of Modern Approach to SCI Security Management and Distribution";
- ABS-ABO-96-SC-002, "Data Networks Vulnerable to Intrusion / Exploitation / Disruption."

The techniques for secret hiding of messages in an otherwise innocent looking carrier message belong to the field of steganography. The purpose of steganography is to conceal the very presence of secret information. Steganography is not a substitute of cryptography. To the contrary, these two fields can tremendously benefit from each other if properly combined. The steganographic algorithm should depend on a secret key so that without the key, it should be impossible not only to extract the hidden message but also to prove its very existence. For obvious security reasons, the key should never be sent with the carrier and should be carrier independent. This is important because if an adversary gets hold of the hiding algorithm, the whole communication channel would be compromised. To make the communication more efficient, the secret information should be compressed before it is hidden in the carrier. This way the amount of information that is to be sent is minimized. Also, it is easier to hide random looking message into the carrier than to hide a message with patterns and regularities. Encrypting the compressed message before hiding is always recommended and provides essentially a double protection.

The steganographer's job is to make the secretly hidden information difficult to detect given the complete knowledge of the algorithm used to embed the information except some secret key[1]. This so-called Kerckhoff's principle is the golden rule of cryptography and is often accepted for steganography as well. In steganography, however, the principle becomes somewhat blurry because the adversary's ability to detect subtle changes in digital data depends on what she knows about the original carrier and its statistical properties. Simply postulating that an eavesdropper (Eve) knows everything there is to know about the carrier is overly optimistic, restrictive, and unachievable in practice. Eve may know, for example, the exact type of the scanner used by Alice and Bob to exchange messages hidden in digital images. Eve can buy the same brand of scanner and build a detailed model of the noise. However, if Eve builds a really detailed model of her scanner noise, she may overdo her job

[1]This is not a uniformly accepted principle, see Tinsley's [26] arguments against Kerckhoff's principle.

and actually detect differences between the two pieces of hardware rather than secret messages. On the other hand, if the source of images Alice and Bob use is not known to Eve, she has to learn the noise by intercepting the communication channel and collect enough information to build a model. However, this task is a difficult one because Alice and Bob can be creative and pick the images randomly from some secret shared database or even modify the noise component using a secret key. Another complication is that it is difficult to estimate the noise model from images containing the noise. The noise may be pixel-dependent and the only reliable way would be to scan each image many times to accurately estimate the noise component.

The field of steganography is very old. Recent steganographic methods include invisible ink and microdots. Microdots are blocks of text or images scaled down to the size of a regular dot in a text. Shifting words and changing spacing between lines in text documents or in images containing text can also be used for hiding bits of secret messages [3–5]. Today, it seems natural to use digital images, digital video, or sound for hiding of secret messages. The gaps in human visual and audio systems can be used for information hiding. For example, the human eye is relatively insensitive to high spatial frequencies. This fact has been utilized in many steganographic algorithms, which modify the least significant bits of gray levels in digital images or digital sound tracks. Human eye is also insensitive to gradual changes in brightness. By overlaying an irregular pattern with gradual changes over the image, the embedded information is incorporated into low frequencies and becomes more robust with respect to common image processing operations and lossy compression. The masking properties of the human visual system can be quantified using mathematical models. Such models can then be applied for design of very robust steganographic techniques with relatively large data capacity [24,25].

Recently, there has been some confusion and misconceptions regarding steganographic methods in connection with digital data. Some authors do not correctly distinguish between invisibility, undetectability and robustness with respect to intentional and unintentional modifications, and the security. By definition, steganographic methods should not visibly modify the carrier. If the carrier is an image, the modifications should be imperceivable to the human visual system, if the carrier is an audio track; the changes should be imperceptible to the human ear. This, however, does not necessarily rule out the possibility that it may be possible to mathematically prove that applying some sophisticated statistical tests has modified the carrier. *The purpose of data hiding can vary, and it is the specific application that imposes different requirements and limitations on the appropriate steganographic methods.* We propose to divide data hiding schemes into two categories based on different requirements the techniques must satisfy.

Secret message hiding

Purpose	To send a secret message without raising any suspicion that a secret message is being sent.
Requirements	Secrecy, difficult to detect, large capacity.
Remarks	Always assume Kerckhoff's principle (protect the embedded information by a message and carrier independent key), maximize the capacity, minimize key size.

Watermarks

Purpose	Copyright protection, proof of ownership, fingerprinting (tracing distributed multiple copies), image integrity protection.
Requirements	Robustness, difficult to remove even under collusion and Kerckhoff's principle.
Remarks	We do not require that the watermark not be detectable in the sense of maximum secrecy as in the case of covert communication

The first category comprises of techniques that hide a large message inside a carrier image. The main requirements are channel capacity and security. A simple example of such a technique is the least significant bit encoding where the least significant bit of each pixel is replaced with one bit of an (encrypted) message. We stress that the hidden message is not required to be robust with respect to modifications of the carrier image. The method is used for covert communication and its primary purpose is to hide the very presence of secret communication. One can imagine a web page containing images, which are periodically updated. Although the messages are available to everybody, only those possessing a secret key can view the hidden information. This way, the very presence of communication is well masked. An important requirement is that it should be computationally difficult to detect the presence of (but not necessarily be able to read) the hidden message.

The second category of message hiding methods has complementary requirements. The main requirement is the robustness of the hidden message with respect to carrier modifications, while the capacity of the hidden channel may be as low as one bit. Such techniques find important applications in digital watermarking for copyright protection, fingerprinting for traitor tracing, image integrity verification, and authentication of digital documents.

1.2 Phase I Technical Objectives

The purpose of this subsection is to review the specific Phase I objective for each task, as outlined in the Phase I proposal, and summarize the results of the Phase I research.

Four primary objectives existed for this research. As outlined previously in the Phase I proposal, these were:

1. Creating schemes for hiding messages in images by slightly modifying the pixel values in an existing image (a carrier) so that a message or simply a different image can be hidden in the carrier.
2. Extending the schemes to images in graphic formats, which utilize lossy compression algorithms. Since a robustness with respect to small amount of noise and / or to loss of information due to lossy compression is necessary, we intend to work in the Fourier / wavelet space instead of the pixel space.
3. Studying the security, efficiency, and robustness of schemes for hiding messages and implementing the algorithms. The security with respect to known attacks will be investigated.
4. Demonstrating the performance of the hiding schemes on real imagery.

Each of the above issues was examined using a theoretical approach and an experimental approach based on computer simulations. In the following sections, we present the detailed results of the Phase I research.

2. Results of Phase I Research

In this section we detail the approach, results, and conclusions with respect to each of the proposed Phase I tasks and objectives. Objectives 1 and 2 as specified above are discussed in Subsections 2.1 and 2.2, respectively. Objectives 3 and 4 include the study of security, efficiency, and robustness of the proposed schemes, and testing on real imagery. These objectives were pursued simultaneously with objectives 1 and 2 and are discussed for each proposed data hiding technique in Sections 2.1–2.3.

In particular, we proved that it is possible to hide messages in the noise component of digital images in such a way that specific statistical characteristics of the carrier image are not disturbed. This way, the hidden message is well masked and its presence cannot be proved even using computers and known statistical measures. We note that it is important to inject the bits of the secret message while taking into account the properties of the carrier image. This adaptive message hiding should be based on local properties of the carrier image. Another aspect of secure steganographic techniques based on digital images is that message bits should be spread pseudo-randomly over the carrier image independently of the message lengths and should not be concentrated in the first rows or columns of the carrier image. For this purpose, a simple but effective message-spreading algorithm based on parameterized chaotic maps was developed. The algorithm uses discretized two-dimensional chaotic maps as universal dispersers for random message scattering within the carrier image.

One of the main contributions of this report is the design of several new secure watermarking schemes for digital images. The new techniques offer a very high degree of robustness with respect to image modifications including noise and lossy compression, and are extremely secure. The first technique is based on overlaying the carrier image with a key-dependent image whose power is concentrated in the low frequencies. The extraction of the hidden message is based on a discrete cosine transform. The second technique inserts bits of

secret messages into projections of image blocks onto random, smooth, orthogonal patterns individually generated for each image and user ID. This way, we avoid using publicly known basis functions, such as discrete cosines, which increases security against malicious attacks.

The robustness and security of message techniques developed in Phase I was tested on real imagery. In particular, we tested robustness with respect to blurring, lossy compression, cropping, resampling, noise adding, and image filtering. We tested the security with respect to the collusion attack (combining multiple images with different embedded messages in order to remove the hidden messages). The algorithms were implemented in Matlab (the source code for all Matlab m-functions is enclosed on a floppy disk).

2.1 Task 1: Non-robust Data Hiding in Images

2.1.1 Using chaotic maps as universal dispersers

Aura [1] proposes the following steganographic method possessing absolute secrecy. To embed a small message of the order of 8 bits or so, just keep scanning an image till a certain password-dependent message digest hash function returns the required 8-tuple of bits. This method has the advantage of absolute secrecy tantamount to one time pad used in cryptography. It guarantees the same error distribution and undetectability. Although the scheme satisfies the requirements of a steganographic holy grail, it is time consuming, has very limited capacity, and is not applicable to image carriers for which we only have one copy.

Aura also suggest to play it safe and change only a small fraction of the carrier bits. For example, modify each hundredth pixel in the carrier by one gray level. Depending on the image noise, these changes will hopefully be compatible with the uncertainties involved with any statistical model of the image. This technique imposes limits on the maximal capacity of the carrier image, and it can leave traces in the carrier image (see the discussion on adaptive message hiding below).

The security of techniques based on modifying the least significant bit can be significantly increased if the consecutive bits of the secret message can be embedded into a pseudo-randomly chosen sequence of pixels of the cover image. While it is certainly possible to design schemes using pseudo-random generators, we propose a scheme in which chaotic permutations [12] are used to randomly scramble the cover image. The secret message consisting of k bits is embedded into the first k pixels of the scrambled image. The inverse chaotic permutation is applied to get the modified cover image. This scheme has the advantage that it is very easily implemented, it is fast, and provides a good security. The scheme is explained in Figure 1.

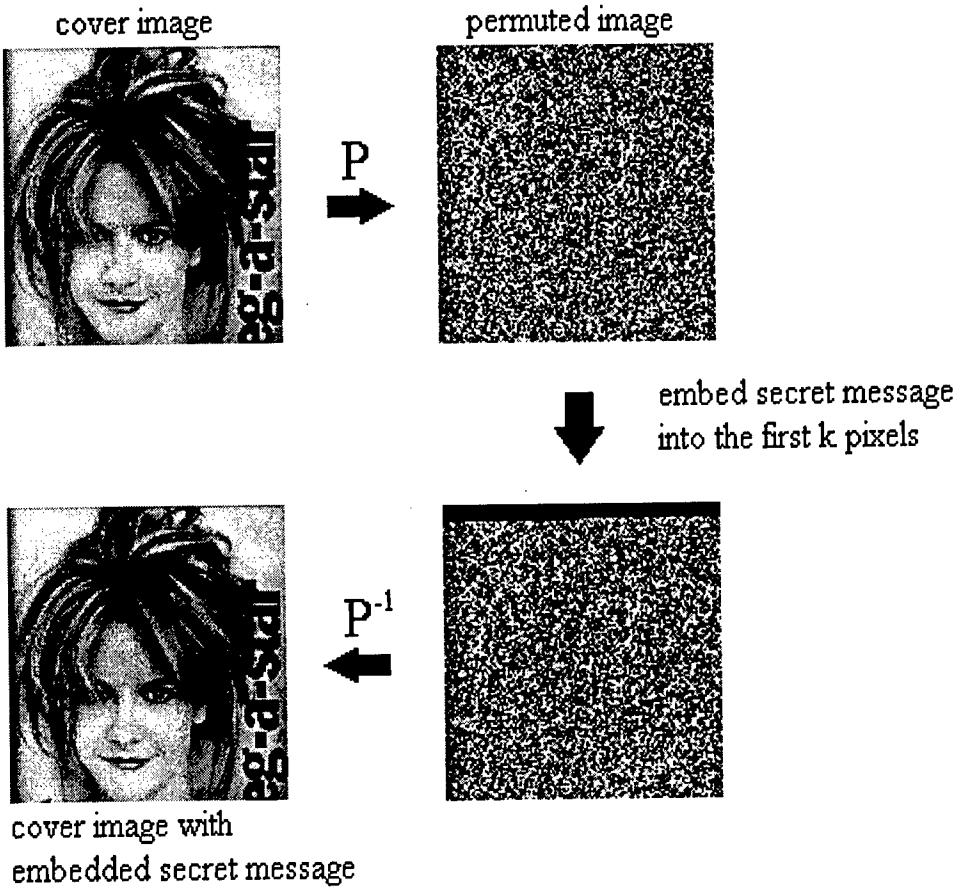


Figure 1 Using chaotic permutations as universal dispersers.

Chaotic permutations generated using two-dimensional chaotic maps, such as the baker map [12], depend on a sequence of integers which add up to N – the number of pixels in one row of the cover image. The permutations are therefore parameterized by the sequence of integers, which play the role of a secret key without which the retrieval of the message becomes impossible.

2.1.2 General methodology for secret message hiding in digital imagery

In this report, we argue that before we can claim any secret message hiding technique as secure, we need to carefully investigate the carrier image and its statistical properties. The noise component may not be uniform within the image but may depend on the pixel position in the image. For example, pixels corresponding to a bright white color will probably be always saturated at 255 even though the overall model of the noise can be Gaussian with a non-zero variance. An example of such an image is shown in Figure 2. The image has been scanned on a scanner. Figure 3 is a black and white image with black pixels corresponding to even gray levels and white pixels corresponding to odd values of gray levels. One can clearly see a large patch of odd pixels, which corresponds to pixels saturated at 255 in the original image. Even if we play it safe and modify only a small fraction of pixels in the image, we will introduce some suspicious noise into the overflowed patch. This problem with over/under

Figure 2 A scanned image.

Figure 3 The same image with white pixels corresponding to odd gray levels and black pixels corresponding to even gray levels.

flow can of course be avoided by a more careful choice of the carrier image, preprocessing the carrier, or by instructing the steganographic scheme to avoid the over/under flowed areas.

While we agree that it is probably impossible to get a complete model of the carrier noise, and that the search for the perfect steganographic method will probably never be complete, we insist that all good secret-hiding schemes must be based on some model of the noise. If it is known that scanned images exhibit larger noise correlations in the horizontal direction and smaller correlations in the vertical direction, while the probability distribution for each pixel, which is neither overflowed nor underflowed, is Gaussian with certain standard deviation, then we have to take this evidence into account and tailor our secret message hiding scheme so that the carrier modifications are consistent with the statistical evidence. It is certainly possible that somebody will, with great effort, create even more sophisticated noise model and detect the presence of messages, but only at the price of painstaking time-consuming and possible expensive investigation. It is rather unfortunate but understandable, however, that most detailed technical information regarding noise in CCD arrays and scanners is proprietary and kept secret by private companies.

To give an example how one can incorporate statistical evidence into the construction of a secret message-hiding scheme is as follows. Let us assume that the noise component of pixels with gray levels within the range [L, H] can be modeled with a uniformly valid probability density, f, symmetric around zero. If the secret plain-text message $\{p_i\}^N_{i=1}$ is encrypted, the cipher-text $\{c_i\}^N_{i=1}$ should be a random sequence of ones and zeros. By averaging several scanned versions of the carrier image, we obtain a "zero noise" image Z. Using a pseudo-random number generator, we can choose at random N pixels in Z with their gray levels in [L, H]. Then, we can modify the LSB of those pixels by the amount of $(2b_i - 1)|\eta_i|$, where η_i is a random variable with probability distribution f. The remainder of the pixels will be modified by η_i. The modifications should be consistent with the statistical model.

2.1.3 Theory of data embedding techniques

A general framework for data hiding schemes is described. The framework encompasses all possible data hiding techniques using one formalism in a compact and elegant manner. If the embedded information is orthogonal to the carrier image, the original unmodified image is not needed for extraction of the secret message. However, modes, which are orthogonal to typical images, are likely to be removed by image compression schemes, such as JPEG or wavelets. On the other hand, messages embedded into the main modes of the image (for example, the lowest frequencies) may create artifacts and the original carrier is needed to extract the hidden message. This can be improved by adjusting the modes selectively according to the properties of the human visual system [9,20,24,25]. The bonus of the latter techniques is that they are extremely robust to modifications. As long as the carrier with the embedded message is not distorted beyond recognition, one can algorithmically prove the existence of the embedded message.

The framework is based on the paper by Comiskey and Smith [6], who give a nice interpretation of data hiding schemes. Carrier image C is modulated by adding a linear combination of basis functions B_i to obtain a stego-image S. The functions C, S, and B_i are defined on a rectangular lattice of points (x, y) and take on integer values (in the case of gray-scale images and palette-based images) or triples of integer values for other color images.

Stego-image (S) = Original carrier image (C) + Modifications (M).

$$\text{Modifications} = \sum_{i=1}^{n} m_i B_i(x, y), \tag{1}$$

where $m_i \in \{-M, -M+1, \ldots, M-1, M\}$, and B_i are defined on the image domain (usually a rectangle).

Security is either guaranteed by keeping the original carrier image C secret or by making the basis functions B_i pseudo-random (key-dependent) or both. We note that if we insist that the original image be kept secret, it becomes a part of a (large) key.

Recovery of the hidden message is made possible either because we know the original image, or by making the basis functions orthogonal to the carrier image. If the original unmodified carrier image is available, we can simply subtract $S - C$ and solve the system of equations for the n unknown m_i :

$$S - C = \sum_{i=1}^{n} m_i B_i(x, y).$$

If the supports of the basis functions are disjoint, no equations need to be solved and the unknown m_i's can be readily recovered. In data hiding schemes without image escrow (without knowing the original image) we must require B_i to be orthogonal to the carrier image

C. Recovery of m_i is then simply achieved by projecting[2] the stego-image S on each basis function B_i

$$\langle S, B_i \rangle = \langle C, B_i \rangle + \sum_{j=1}^{n} m_j \langle B_j, B_i \rangle \cong \sum_{j=1}^{n} m_j \langle B_j, B_i \rangle.$$

In order to recover the hidden message, we need to solve this system of n equations. If we make the basis functions B_i orthogonal to each other, no equations need to be solved and we have an estimate $\bar{m}_i$ for m_i

$$\bar{m}_i = \frac{\langle S, B_i \rangle}{\langle B_i, B_i \rangle}.$$

This estimate will not be exactly equal to m_i for the following reasons. First, the basis functions will not be generally exactly orthogonal to the carrier C unless we design them so. If we make them exactly orthogonal, they will be carrier-dependent and thus become part of the key. Second the stego-image S may be intentionally or unintentionally manipulated. In order to make the recovery process more robust, we suggest to calculate all the estimates $\bar{m}_i$ and threshold them to the discrete values $-M$, $-M + 1$, ..., $M - 1$, M. In most data hiding schemes, the values m_i are of binary nature alternating between $-G$ and G without taking any intermediate values. Comiskey and Smith [6] simply calculate all the estimates $\bar{m}_i$, $i = 1,\ldots,$ n, and threshold them with respect to the median of their maximum and minimum values. However, as they note, for this to work we need to make sure that not all m_i are the same.

Functions B_i can be spatially localized or localized in frequency space, or localized in both spatial and frequency domain. They can also be spread in space or frequency. Different choices of the basis functions lead to different data hiding schemes. For example, the simplest implementation of LSB encoding corresponds to B_i 's with a one-pixel support and $M = 1$.

2.2 Task 2: Robust Data Embedding Schemes (Watermarking)

Some applications may require the secret data to be embedded in a robust way so that small changes to the carrier image do not destroy the hidden data. The carrier image may be subjected to a lossy compression scheme, such as JPEG; it can be manipulated in an image processing software. Clearly, the higher the robustness the smaller the size of the message that can be hidden inside the image. An important instance of robust message hiding is transparent watermarking.

The most important requirement for digital watermarks is their robustness with respect to common image processing procedures, such as kernel filtering, lossy compression, digital-analog conversion, resampling, cropping, affine transformations, removing / adding features,

---

[2]The dot product is defined in the usual way as $\langle B_i, B_j \rangle = \sum_{x,y} B_i(x,y) B_j(x,y)$, where (x, y) runs through all pixels in the image.

noise adding, copying, and scanning. The second important property of digital watermarks is their non-removability. This could be achieved by making the algorithm for watermark embedding cryptographically strong using a secret key. The purpose of a digital watermark is to prove the ownership of digital data. Robustness with respect to these operations can only be achieved at the price of making detectable (i.e., not subtle) changes to the image. It is probably impossible to create an undetectable (in steganographically secure sense) watermark, which is robust.

If we can design an algorithm for embedding of robust watermarks, the watermarks can be used for solving a dispute about an ownership of some digital data. Let us say that A creates a digital image and watermarks it with his key. If B gets hold of the watermark image and wants to steal the image (i.e., claim the ownership), the best he can do is to embed his watermark into the image and claim that he can prove the ownership of the image. If the watermark is robust, it is easy for A to prove the ownership to a judge because A can detect his watermark in the image marked by B, while B cannot do that for the image that A has. Since we trust the watermarking algorithm that it cannot be completely removed unless one modifies the image beyond recognition, this proves the ownership of the image. We repeat that the watermark's robustness with respect to changes plays a crucial role in this dispute because if B can filter out the original watermark, the watermark would be useless.

It is important that the watermark depend on the original image, otherwise a forgery could be constructed by simply creating an arbitrary watermark and subtracting it from the watermarked image to get a forged original. This so called IBM attack [8] can be prevented by making the watermark pattern depend on the original image in a non-invertible manner. Hash functions are usually employed in these tasks [8, 11].

What does it mean to prove that watermark is inside an image? It is to show that certain relationship among pixels (the reconstructed watermark) can only occur by a pure chance with an extremely low probability.

Another purpose of data embedding is to mark images so that large modifications of the image, such as removing and/or adding objects can be detected. In this case, the watermark would carry some information about the features present in the image. For example, we may robustly encode values of several "hash functions" of the original image. The word hash is in quotes because we actually do not want an absolute sensitivity with respect to the image data, but some other functions, which would be non-collision in some robust sense. For example, we could take a correlation of the original image with one or more secret masks. While this would be effective against removing / adding of features, it may not work as well for, say exchanging two small human faces in a photograph.

2.2.1 Watermarking Method I

We propose new robust watermarking techniques to remove the following weakness in the scheme introduced by Cox et al. [7]. Their watermarking method can be interpreted as overlaying a pattern, which is formed by a linear combination of discrete cosines. The watermark values are used directly as coefficients of that linear combination. The watermark may become visible (or at least detectable) in those areas of the carrier image which were

originally uniform or had a uniform brightness gradient[3]. The watermark cannot be readily removed because the discrete cosines do not generally form a set of linearly independent functions on proper subsets of the image. Nevertheless, this situation is potentially dangerous and could be used to mount an attack on the watermark. Let us assume that a square area containing K pixels reveals some approximation to the watermark. Since we know that the watermark is spanned by the lowest 1000 coefficients, we can solve K equations for the 1000 unknown DCT coefficients A_r and narrow down the possibilities for the watermark sequence by a large margin:

$$g_{ij} = \sum_{r=1}^{1000} A_r \cos\left(\frac{\pi f_{1r}(2i+1)}{2N}\right)\cos\left(\frac{\pi f_{2r}(2j+1)}{2N}\right), \quad (i,j) \in \text{Area consisting of K pixels.}$$

In the equation above, $f_r = (f_{1r}, f_{2r})$ denotes the r-th 2d frequency, and A_r denotes the r-th unknown coefficient. In Section 2.3, the attack is explained in more detail and a simplified version of the scheme [7] is successfully attacked.

There are at least three ways to prevent the attack. One possibility is to use adaptive watermarks whose strength is locally adjusted according to the masking properties of the human visual system. This will be further studied in Phase II of this project.

The second possibility is to replace discrete cosines with other, key-dependent bases. If the basis functions are not known, this type of attack would not be possible. In order to make this practical, however, we would have to design orthogonal bases which would depend on parameters – a secret key. Another important requirement is that there should exist efficient computational algorithms similar to fast Fourier type of transforms. An algorithm based on orthogonal, key-dependent, smooth patterns (basis functions) is presented in Section 2.2.3.

The third option is to view the watermarking scheme as pattern overlaying. We do not have to use patterns formed by a linear combination of discrete cosines but we could utilize general key-dependent patterns with their power concentrated mostly into low frequencies in order to guarantee robustness. This approach is further elaborated below.

We start with a procedure that transforms a string of bits (a watermark) into a smooth, almost transparent pattern to be overlaid over the carrier image. We require a sensitive dependence between the watermark and the resulting pattern. The pattern should not exhibit traces of any regular building blocks. Also, patterns generated by two different watermarks should be uncorrelated. While we realize that there probably are many ways how to achieve these goals, we propose to seed a random number generator with the watermark to create an initial black and white two-dimensional random pattern. The pattern will be further processed to eliminate high frequencies. This can be done for example by applying low-pass filters to the initial pattern. Another possibility would be to use the initial pattern as an initial condition for some chaotic spatio-temporal dynamical system, which has the tendency to create large-scale smooth structures. Still another possibility would be to use cellular automata with appropriate rules. For example, it is known that voting rules and their modifications [27] can coalesce random patterns into large-scale structures.

---

[3]Quite a large percentage of images do contain such areas.

In Figure 4a we show an example of a randomly generated initial pattern. Since the purpose of this study is to show the plausibility of the proposed watermarking scheme, we used a simple random number generator supplied with Matlab V. The initial pattern was initialized with 0's and 1's with the same probability (50%). The random black and white pattern was then processed by a cellular automaton with the voting rule [27]. According to this rule the center element follows the majority of its neighbors. In particular, we count the number, P, of 1's in the 3×3 neighborhood of the center (including the center) and set the center to 0 if $P < 5$, otherwise we set the center to 1. The cellular automaton with voting rules always stops after finitely many steps (for a 128×128 image, we found this number to be always less than 40). The resulting pattern together with three intermediate patterns is shown in Figure 4b – Figure 4e. As can be seen, the random pattern has coalesced into several connected areas forming an irregular pattern which depends on the seed of the random number generator (and therefore on the watermark sequence) in a complicated way. As the last step, we applied a smoothing operation with a 7×7 kernel eight times. The color depth of the resulting image was decreased to 16. Examples of five different patterns are shown in Figure 5. Before we add these patterns to an image, we subtract 8 from each pixel value of the pattern so that the pixels of the original unwatermarked image are modified by no more than ±8 gray levels.

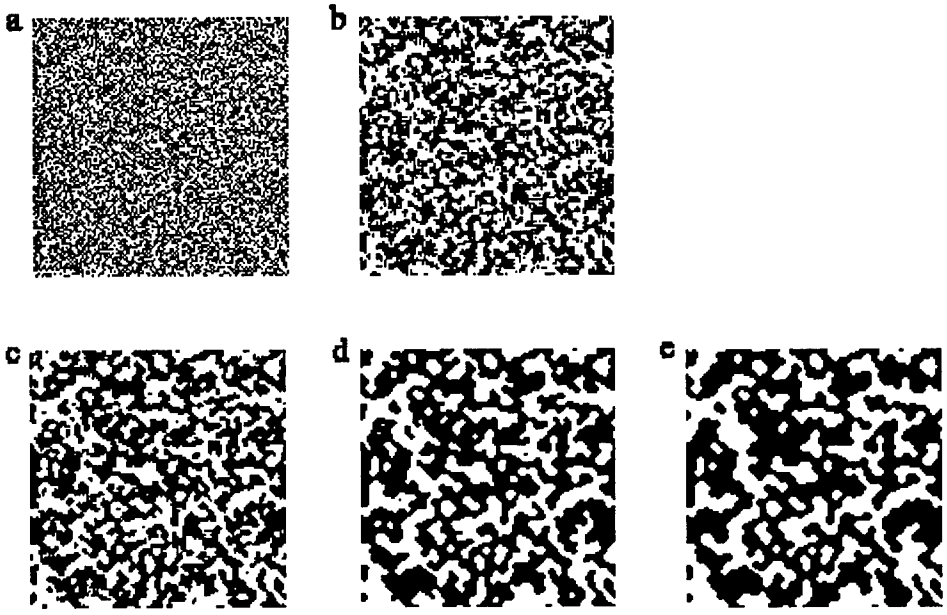


Figure 4 The initial pattern a) after 1, 2, 5, and 35 iterations of a cellular automaton with voting rules.

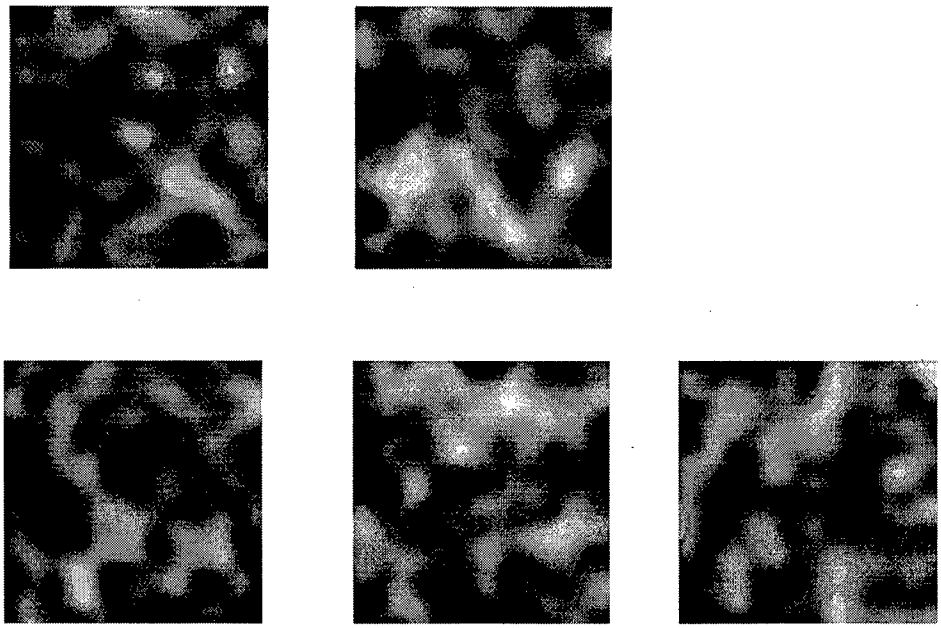


Figure 5 Examples of five watermarking patterns. The contrast was enhanced to make the patterns visible.

Since the overlaid image has most of its power concentrated in low frequencies, we expect excellent robustness properties similar to the method proposed by Cox et al. [7]. Since our overlaid pattern depends on the key in a complicated way, even if the watermark shows in regions of nearly constant luminosity, it does not reveal any information about the key if a cryptographically strong random number generator is used. Another advantage of this technique is that it avoids transformations, which results in a faster and easy implementation.

Our new scheme for embedding digital watermark data can be described as follows: Watermark sequence → seed for a random number generator → initial two-dimensional black and white random pattern → cellular automaton → smoothing algorithm → adding the pattern to the image.

Watermark retrieval: Subtract the watermarked image from the original and calculate the correlation between the difference and the smoothed pattern. Based on the value of the correlation, decide about the presence of the watermark. In our experiments, the correlation in the Fourier space performed better than in the spatial domain. Denoting the original unwatermarked image as X, the watermarked image X', the modified watermarked image X^* the following function was used to decide about the presence of a watermark:

$$sim(X^*, X') = \frac{D^* \cdot D}{\sqrt{D^* \cdot D^*}\sqrt{D \cdot D}} \quad ,$$

where $D = Y' - Y$, $D^* = Y^* - Y$, and the symbols Y, Y', and Y^* denote the lowest 1024 discrete cosine coefficients corresponding to X, X', and X^*, respectively.

Since it appears rather difficult to analyze the mechanism how cellular automata coalesce randomly scattered points into connected clusters; we base our analysis on a series of

computer experiments only. Our conclusions regarding the robustness of the method are based solely on computer experiments. Clearly, this is not satisfactory, and a more exact analysis of the method needs to be done. This will be an important part of the future research.

We determined experimentally that for images X and Y watermarked by a different watermark, $sim(X, Y)$ is mostly confined to the interval [–0.05, 0.05] with the maximal observed value of 0.175.

Robustness with respect to JPEG compression. The carrier image with 128×128 pixels and 256 gray levels and its watermarked version are shown in Figure 6. The marked image has been compressed using JPEG compression method with quality factors 50%, 20%, 10%, and 5%. The compressed images are shown in Figure 7.

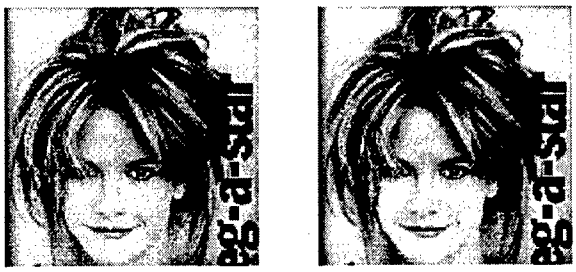


Figure 6 The original test image (left) and a watermarked copy (right).

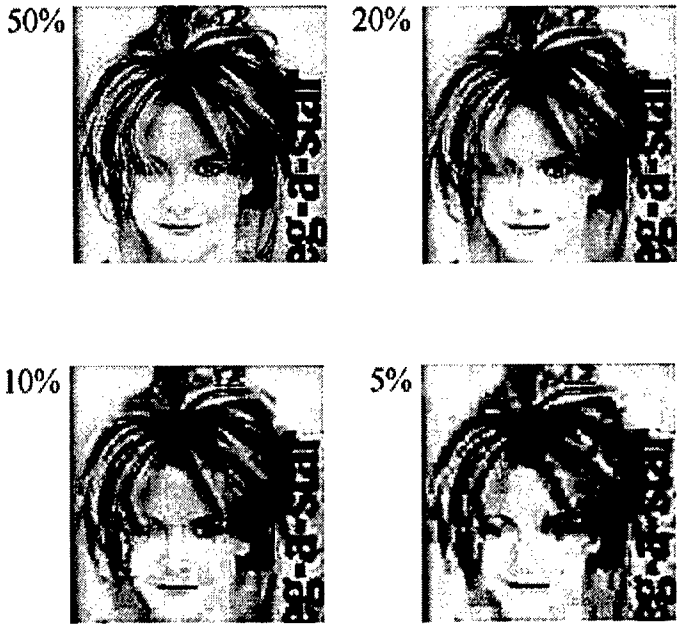


Figure 7 The watermarked image JPEG compressed with 50%, 20%, 10%, and 5% quality factor.

As can be seen, the quality factor 5% severely distorts the original image. Nevertheless, the watermark can still be detected. The corresponding values of *sim* are shown in the table below.

Table 1 Robustness of Method I to JPEG encoding.

Quality factor	50%	20%	10%	5%
sim	0.97	0.84	0.62	0.38

Table 1 clearly shows that common JPEG compression with quality factor equal to 50% almost does not disturb the watermark. As low quality factor as 5% still leaves the watermark detectable because the value of *sim* = 0.38 is twice as much as the largest similarity for two random watermarks.

Figure 8 Test image after downsampling by a factor of 2 and upsampling.

Robustness to mosaic filtering. The test image 128×128 has been downsampled to a smaller image 64×64 and then upsampled again to the previous size (see Figure 8). Although 75% of information has been lost in this process, the value of *sim* for the mosaic-filtered image was 0.44, which indicates the presence of the watermark.

Robustness to blurring. We used PaintShop Pro and its "blur more" filter to blur the test image. The resulting blurred image is shown in Figure 9. The similarity number 0.43 shows the presence of the watermark.

Figure 9 Blurred image.

Robustness to median filtering. PaintShop's median filter was used to create Figure 10. The similarity value for the filtered image was 0.71. This value indicates the presence of the watermark.

Figure 10 Median filtered test image.

Robustness to cropping. Figure 11 shows a white rectangular area which has been cropped from a watermarked image. The cropped watermarked area forms about 23% of the total image area. The missing part of the cropped image was replaced with portions of the original, unwatermarked image. The similarity function applied to that image returned the value 0.48. This shows that even though more than 77% of image information has been discarded due to cropping, the watermark remains detectable.

Figure 11 The white rectangle indicates the cropped area.

Robustness to uniform noise. We added 30% of uniform noise into the test image (see Figure 12) and tested for presence of the watermark. The similarity value of 0.57 clearly indicates the watermark's presence.

Figure 12 The test image after adding 30% of uniform noise.

Robustness with respect to collusion attack. In this test, we created 5 different watermarks and averaged the watermarked images into a single image. The (enhanced) watermark patterns are shown in Figure 5. Then, we tested the averaged image for presence of the five watermarks. The similarity values are shown in the table below.

Table 2 Robustness of Method I to collusion.

Watermark #	*w*1	*w*2	*w*3	*w*4	*w*5
sim	0.42	0.44	0.48	0.56	0.44

In an attempt to develop an embedding scheme, which would share both the robustness and the convenience of not requiring the original carrier image for message extraction, a new data embedding technique was developed. It is based on creating a robust relationship among selected coefficients of low-frequency discrete cosine modes. Testing of the method will be a part of the future effort. We also tested the averaged image for presence of 20 randomly generated watermarks. In neither case was the value of *sim* greater than 0.17.

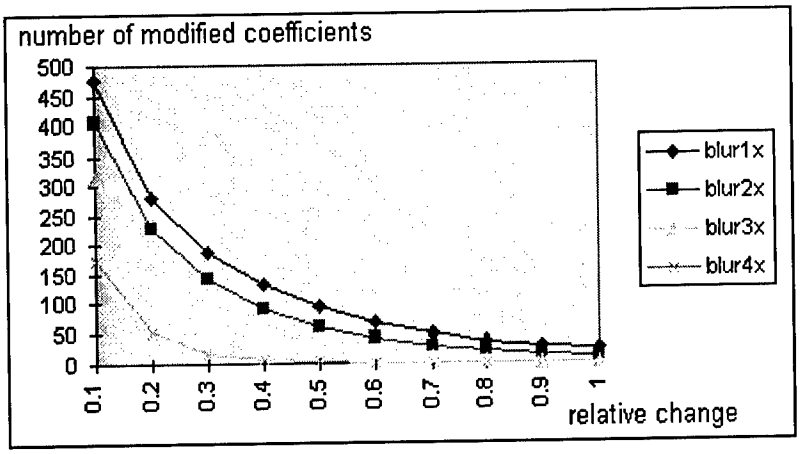
2.2.2 Watermarking Method II

Watermarking techniques that do not require the original image in order to extract the hidden data have numerous applications, such as automatic detection of stolen images, intelligent Internet browsers capable of filtering out inappropriate information to a particular user, etc. The schemes usually create some correlations in the image such that the probability of occurrence of those correlations by pure chance is very small. Bender's texture block mapping technique [2] achieves that by cropping a random looking area in the image and pasting it in some other random looking area. However, this watermark can be readily detected and removed. We need to create correlations, which would be protected by a secret password. We choose to work in the Fourier space and make modifications to the coefficients of discrete cosine transform rather than to the pixels.

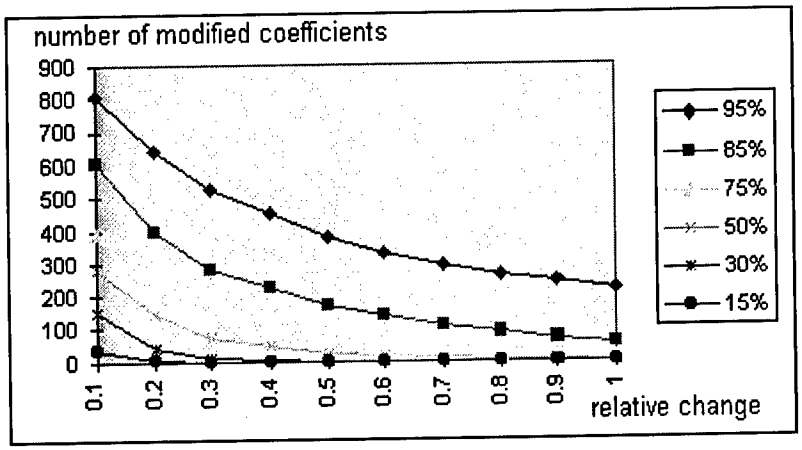
The technique utilizes a robust relationship among randomly chosen lowest frequency coefficients of the discrete cosine transform (DCT). The coefficients are sequentially modified to encode a specific binary pattern. Detection of the watermark consists of checking the pseudo-random sequence of coefficients for the desired pattern. Hence, the original image is not needed for extraction of the watermark.

Prior to designing the watermarking method, we performed an analysis of how certain image processing operations influence the lowest DCT coefficients. These tests will tell us how and by how much the DCT coefficients should be changed. Three operations were tested: blurring, downsampling, and lossy JPEG compression. The results are shown in the diagrams below. The legend for all three diagrams is the same. The horizontal axis x is the size of the relative change of the coefficient (a value of 0.2 means a 20% change). On the vertical axis y, there is the number of coefficients out of 1000 lowest frequencies, which have been modified by at least x.

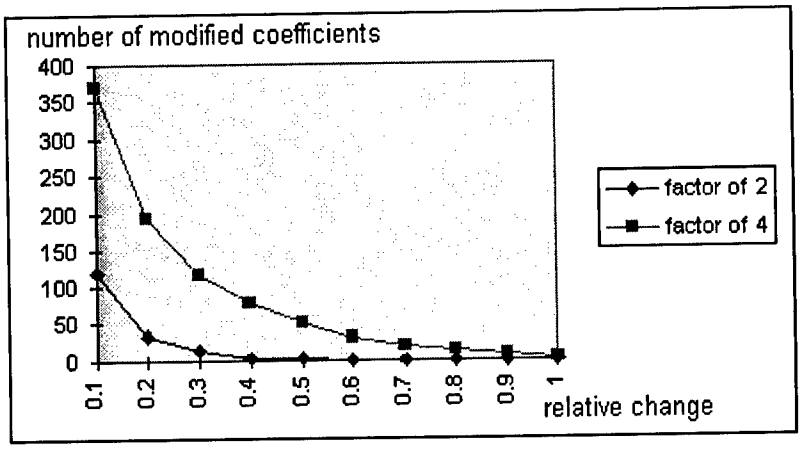
Blurring



JPEG compression



Downsampling



By comparing the figures for lossy JPEG compression, one can clearly see that the relative percentage of modified coefficients quickly grows with the increasing compression ratio. Since a 50% quality factor generates acceptable images, we have to increase the watermark strength in order to keep a certain acceptable level of robustness.

The proposed method is based on the scheme proposed by Cox et al [7]. For $0 < \alpha < 1$ we define the sequence

$$\{x_i\}_i,\ x_{i+1} = \frac{1+\alpha}{1-\alpha} x_i,\ i = -\infty, \Lambda\ , \infty.$$

This is a geometric sequence with quotient $q = (1+\alpha)/(1-\alpha)$, $q > 1$. Since the sequence is strictly increasing, with x_i approaching zero as i goes to $-\infty$ and x_i diverging to ∞ with increasing index i, any positive real number x belongs to exactly one interval $x_i \le x < x_{i+1}$. The value of the quotient q guarantees that any real number x can be modified by at most 100α % in order to move x to some other interval $[x_i, x_{i+1})$. This can be reformulated using an index function $ind(x)$

$$ind(x) = 1 \quad \text{if } x_i \le x < x_{i+1},\ i \text{ even}$$
$$ind(x) = 0 \quad \text{if } x_i \le x < x_{i+1},\ i \text{ odd}.$$

Any x can be modified by adding or subtracting at most αx to change its index $ind(x)$.

To embed a watermark, the image is transformed using a discrete cosine transform. Originally, we have experimented with the following design: Pseudo-randomly choose k out of N lowest frequency coefficients v_i and modify them by $\pm\alpha w_i$ so that a simple binary pattern is encoded

$$ind(v'_i) = (-1)^i,\ v' = v_i(1+\alpha w_i),\ i = 1, \mathrm{K}\ , k.$$

Perform these modifications in a specific order determined by a pseudo-random generator (*the order in which the N coefficients are chosen is important*). The properties of the sequence $\{x_i\}_i$ guarantee that the required changes will never have to be greater than $\pm\alpha v_i$. This design, however, has a flaw. In the cases when v_i is close to the midpoint m_i , the modified value will be close to the end point of the index interval. This will make the watermark undesirably sensitive to noise adding. The robustness could be improved if the modified value v'_i was always close to the midpoint of the index intervals. This would, however, require a slightly larger modification of the coefficient v_i. It can be argued that the modification will never have to be greater than $(1+\alpha)/(1-\alpha)v_i$. This corresponds to a relative change of

$$\frac{1+\alpha}{1-\alpha} = 1 + \frac{2\alpha}{1-\alpha} \approx 1 + 2\alpha\,.$$

We can also say that any v_i will be modified by at most 200α %. The complete description of the algorithm is given below.

The value of x_0 and α are kept secret as part of the author's key. Since the value of α is secret, we can afford to modify the DCT coefficients so that they stay in the middle of the index intervals. The bit sequence $ind(v_i)$ can be public or secret. We choose the lowest N coefficients of the DCT and pseudo-randomly choose $k<N$ coefficients. The modifications of the DCT coefficients were done in the following way. First, we calculated the sequence $\{x_i\}$ by setting $x_0 = 1$. For each coefficient c, find the integer r, such that $x_r \le c < x_r\,(1+\alpha)/(1-\alpha)$. Compare c with the midpoint $m_r = x_r/(1-\alpha)$. If $c < m_r$ then c is changed to m_{r-1}, otherwise it

is changed to m_{r+1}. Of course, this modification is only necessary in 50% of all cases (in roughly 50% cases, the index of c will be correct). The pseudo-code for this algorithm is as follows.

(1) Take the original image and apply a message digest function to it.

(2) Append author's ID bit-string to the hash.

(3) Use this bit-string to initialize a cryptographically strong pseudo-random number generator.

(4) Pseudo-randomly choose k out of N lowest frequency coefficients of the DCT.

(5) Choose the bit-sequence $ind(v_i)$. This should be done such that the number of 0's and 1' in the sequence is the same (so that the total energy of the image approximately stays the same) and the 0's and 1' should be distributed in some regular fashion. For all of our simulations, we chose this bit-string to be an alternating sequence $\frac{1}{2}((-1)^i+1)$ = 1010101010101....

(6) For $i = 1, \ldots, k$,

determine a positive integer r such that $x_r \le v_i < x_{r+1}$ (it can be shown that $r = \lfloor \log|v_i| / \log q \rfloor$, where $\lfloor z \rfloor$ denotes the integer part of z)

compute the "midpoint" of the r-th interval $m_r = x_r / (1-\alpha) = x_{r+1} / (1+\alpha)$

if $\frac{1}{2}((-1)^i+1) \neq \frac{1}{2}((-1)^r+1)$ *and* $v_i > m_r$ **then**

$\{v_i \rightarrow x_{r-1} / (1-\alpha)$, the midpoint of the interval on the left of the r-th interval$\}$

if $\frac{1}{2}((-1)^i+1) \neq \frac{1}{2}((-1)^r+1)$ *and* $v_i \le m_r$ **then**

$\{v_i \rightarrow x_{r+1}/(1-\alpha)$, the midpoint of the interval on the right of the r-th interval$\}$

if $\frac{1}{2}((-1)^i+1) = \frac{1}{2}((-1)^r+1)$ **then**

$\{v_i \rightarrow x_r / (1-\alpha)$, the midpoint of the r-th interval$\}$

The Matlab code for this algorithm is

```
function Y=mark(X,ind,alpha);

DX=dct2(X);
m=1;
DXX=DX;

[k,l]=size(ind);
coef=(1+alpha)/(1-alpha);
```

```
logcoef=log(coef);
for i=1:k
  m=m*(-1);
  v=floor(log(abs(DX(ind(i,1),ind(i,2))))/logcoef);
  aux=coef^v/(1-alpha);
  if mod(v,2)~=(1+m)/2
    if aux > abs(DX(ind(i,1),ind(i,2)))
     DXX(ind(i,1),ind(i,2))=sign(DX(ind(i,1),ind(i,2)))*(
     coef^(v+1))/(1-alpha);
  else
     DXX(ind(i,1),ind(i,2))=sign(DX(ind(i,1),ind(i,2)))*(
     coef^(v-1))/(1-alpha);
    end
 else
     DXX(ind(i,1),ind(i,2))=sign(DX(ind(i,1),ind(i,2)))*(
     coef^v)/(1-alpha);
  end
end

Y=idct2(DXX);
```

The function mark() accepts three parameters: X, ind, and alpha. The symbol X is a matrix of integers – the image to be watermarked, ind is a $k\times2$ array of integers – (ind(i,1), ind(i,2)) are pseudo-random indexes of k DCT coefficients, and alpha corresponds to α. The output of this function Y is the watermarked image.

To detect the watermark, we calculate the correlation between the calculated sequence $ind(v_i)$ and $\frac{1}{2}((-1)^i+1)$. The Matlab code for the watermark testing function is

```
function y=getwat(X,ind,alpha);

[k,l]=size(ind);
m=1;
s=0;
logcoef=log((1+alpha)/(1-alpha));
```

```
DX=dct2(X);

for i=1:k
 m=m*(-1);
 v=floor(log(abs(DX(ind(i,1),ind(i,2))))/logcoef);
 s=s+(2*mod(v,2)-1)*m;
end
y=s/k;
```

The new technique was tested on real images. Robustness with respect to JPEG encoding, resampling, blurring, and noise adding was tested. We have experimented with N=100 and k=20. A more secure implementation can be obtained by choosing N equal to at least 1000. The robustness results are summarized in the table below.

Table 3 Robustness of Method II to image processing operations.

Operation	Marked image (correlation)	Unmarked image (correlation)
JPEG 50%	0.8	0.3
JPEG 75%	0.6	0.2
JPEG 85%	0.6	0.1
JPEG 95%	0.8	0.1
Blur 1x	0.6	0.3
Blur 2x	0.6	0.1
Resample 2x	0.7	0.4
Noise Adding	0.6	0.1

The security of the technique is guaranteed by the pseudo-random choice of the k coefficients out of N lowest DCT coefficients. Without the knowledge of the key, the parameter α, and x_0 one can blindly try to modify a large number of DCT coefficients by some large value. This will, however, inevitably cause severe distortion of the image. If one could somehow find out the value of α, it might be plausible to detect most of the k pseudo-randomly chosen coefficients. However, direct search for the coefficient α is tremendously computationally intensive. One would have to search through

$$\binom{N}{k}$$

possibilities. For N=1000 and k=50, this number is larger than 10^{98}. Masking the watermark pattern with some perceptual image dependent mask could help alleviate the problem of creating artifacts in smooth areas of the image.

The robustness of watermarking techniques that do not require the original image for watermark detection is understandably lower than for techniques that can subtract the original image before detection, such as the watermarking methods I and III. However, the correlation values as shown in Table 3 are not sufficiently well separated for a reliable scheme. For example, the values after resampling for watermarked and unwatermarked image are 0.7 and 0.4, respectively. This indicates a possible problem with false detections. We conclude that this watermarking technique needs further study before it can be implemented for further study.

2.2.3 Watermarking Method III

One possibility would be to generate a random rotation matrix in a J-dimensional space (where J is a large number of the order of at least 100) and rotate J pseudo-randomly chosen discrete cosines or wavelets. This way, a new orthogonal basis will be obtained.

The high robustness of the method of Cox et al. [7] is due to the fact that the watermark is placed into the most perceptive Fourier modes of the image. The fact that a publicly known transformation is used can potentially become dangerous if portions of the original unwatermarked image can be guessed. The security of the scheme and its versatility could be increased if a different set of orthogonal basis functions would be used for each image. Since the basis functions or, equivalently, the key for their generation, will be kept secret, the watermark pattern could be spanned by a smaller number of functions thus enabling us to embed more bits in a robust and secure manner. To achieve this goal, we need to generate a set of orthogonal random functions (patterns) that sensitively depend on each bit of the initial bit-string formed by a combination of user's ID and an image hash. To guarantee good robustness properties, the generated patterns should have their energy concentrated mainly in low frequencies.

We start our study with a simple observation. Namely, it is not necessary to have a *complete* set of orthogonal basis functions since only a relatively small number of them are needed to span a watermark pattern. One can calculate the dot products[4] of the original image with a set of J orthogonal functions, and modify the dot products so that secret information is encoded. Let us denote such functions f_i , i = 1, ..., J. Assuming that the functions are orthogonal to each other, the system of J functions can be completed by MN–J functions g_i to a complete orthogonal system. The original image I can then be written as

---

[4]The dot product of two images A_{ij} and B_{ij} is defined as $\langle A,B\rangle = \sum_{i,j=1}^{N} A_{ij}B_{ij}$

$$I = \sum_{i=1}^{J} c_i f_i + g, \qquad c_i = < f_i, I >,$$

where g is a linear combination of functions g_i that are orthogonal to f_i. The watermarking process is achieved by modifying the coefficients c_i. Furthermore, the watermarked image Iw can be expressed as

$$Iw = \sum_{i=1}^{J} c'_i f_i + g, \qquad c_i' = < f_i, Iw > = (1+\alpha_i)c_i,$$

where c_i' are the modified coefficients. Given a modified watermarked image Im,

$$Im = \sum_{i=1}^{J} c_i'' f_i + g', \qquad c_i'' = < f_i, Im >,$$

we can calculate the modified coefficients by evaluating the dot products of Im with functions f_i. A cross-correlation $corr$ of the differences $c''-c$ with $c'-c$,

$$corr = \frac{(c''-c)(c'-c)}{\|c''-c\| \, \|c'-c\|},$$

is compared to a threshold to decide about the presence of a watermark.

We now describe a procedure for producing orthogonal functions f_i that have power concentrated in low frequencies, and that depend sensitively on each bit of the secret bit-string – the combination of author's ID and an image hash. The secret bit-string is used as a seed for a cryptographically strong pseudo-random number generator and a set of J pseudo-random black and white patterns are generated[5]. The patterns are smoothened by applying a low-pass filter to them. To make the patterns orthogonal, the Gramm-Schmidt orthogonalization procedure is applied. Finally, the functions are normalized to get an orthonormal system. This way, we obtain a set of J orthonormal functions that have their power concentrated in low frequencies. Moreover, the functions (patterns) sensitively depend on each bit of the secret bit-string.

Our new scheme for embedding digital watermark data can be described as follows: author's ID + image digest → (pseudo-random number generator + smoothing) → a set of J random, smooth patterns → (Gramm-Schmidt orthogonalization process) → a set of J orthonormal, random, smooth patterns → (modifying the dot products) → watermarked image. In our scheme, the first coefficient plays the role similar to the DC term in a DCT. To preserve the energy of the watermarked image, the first coefficient c_1 is left unmodified. The other orthogonal patterns have approximately zero mean.

To retrieve the watermark, calculate the dot products c_i'' with the J secret functions f_i. Compare the coefficients c_i'' with those of the watermarked image and the original unwatermarked image by calculating a correlation. Based on the value of this correlation, decide whether or not a watermark is present. To avoid large memory requirements to store

---

[5]In order to keep the computational requirements at a reasonably low level, we removed the coalescing part based on cellular automaton from the scheme.

all orthogonal patterns, only the image hash and author's ID need to be stored. The orthogonal patterns can be generated for each detection attempt.

A pseudo-code for the watermarking algorithm (gray scale $N \times N$ images):

begin_algorithm

read image $\boldsymbol{I}$; // $\boldsymbol{I}$ is a matrix of integers 0, …, 255

convert $\boldsymbol{I}$ *to an intensity matrix* $\boldsymbol{X}$; // $x_{ij} \in [0,1]$

// Concatenate author's ID and image hash

initial_bitstring = Author's ID ^ Hash of $\boldsymbol{I}$;

// Initialize a PRNG with a hash of the initial bit-string

seed = Hash(initial_bitstring);

Step 1 (Generate J pseudo-random binary patterns and smooth them)

for k=1 *to* J

using the PRNG, generate an $N \times N$ *binary pattern* $Z^k = Z^k_{ij}$, $1 \le i, j \le N$;

$Z^k = smooth(Z^k)$;

end_for

Step 2 (Orthogonalize the smoothened patterns using Gramm-Schmidt orthogonalization procedure)

for k=1 *to* J

$$Z^k = Z^k - \sum_{s=1}^{k-1} < Z^s, Z^k > Z^s$$

$$Z^k = \frac{Z^k}{\|Z^k\|}$$

end_for

Step 3 (Calculate the dot products and modify them to embed a watermark)

for k=1 *to* J

$$c_k = < Z^k, X >$$

$$c_k' = c_k[1 + \alpha(k)]$$

end_for

Step 4 (Calculate the watermarked image *Iw*)

$$Xw = X + \sum_{j=2}^{J} \alpha(k) c_k f_k$$

*Convert Xw to a gray-scale image **Iw**;*

end_algorithm

The coefficients $\alpha(k)$ determine the visibility of the watermark and its robustness. The watermarking scheme could be applied either globally to the whole image, or locally. In the global scheme, the support of the functions f_i is the whole image. This makes the scheme computationally expensive with large memory requirements. For an $N \times N$ gray-scale image, one needs JN^2 bytes to store all J orthogonal patterns. This number could become prohibitively large with even for moderate values of N (such as $N = 256$). The most time consuming part of the algorithm is the Gramm-Schmidt orthogonalization process. It has computational complexity $O(J^2N^2)$. Thus, the choice of the number of orthogonal patterns, J, turns out to be critical. If J is chosen too small, the correlation used for detection of watermarks can have occasionally large values due to random correlations. If J is chosen too large (of the order of 1000 or larger), the computational complexity of the scheme becomes unreasonably large. A good compromise is to break the image into smaller subregions that are watermarked separately using different sets of orthogonal patterns and average the correlations from multiple subregions. The averaging will decrease the values of random correlations, while keeping the robustness sufficiently high and at reasonable computational requirements.

Test 1: Global scheme

Even though the global scheme is not suitable for practical use due to the immense computational and memory requirements, we nevertheless performed tests on a 128×128 square image downsampled to 64×64 pixels (see Figure 13).

Figure 13 Original image.

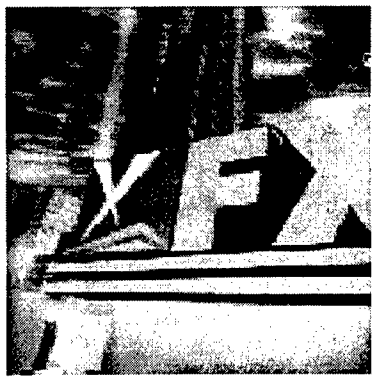


Figure 14 Watermarked image.

Table 4 Robustness with respect to image modifications.

Image operation	Correlation
Blurring (in PaintShop Pro 4.12)	0.75
16% uniform noise (as in PSP 4.12)	0.95
Downsampling by a factor of 2	0.92
StirMark applied once	0.80
Unzign12 applied once	0.82

A 64×64 gray-scale image was watermarked (Figure 14) using 100 orthogonal patterns (the watermark length $J = 100$). It was then tested for presence of 100 randomly generated watermarks. The coefficients $\alpha(k)$ were chosen for simplicity as $0.15(-1)^k$. The correlation is shown in Figure 15. The robustness with respect to JPEG compression was tested for quality factors from as low as 5% to 85% and is shown in Figure 16. The robustness with respect to other image processing operations is summarized in Table 4.

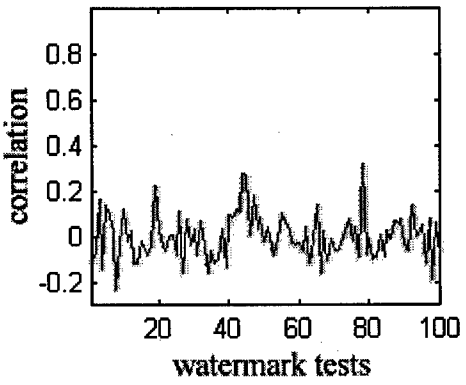


Figure 15 Correlation for 100 random watermarks for global Method III.

The image modifications were done using commercial, PaintShop Pro 4.12 software. Since Unzign12 cut the horizontal dimension of the image by 3 pixel values, the corresponding portion of the original image was used to bring the dimensions back to those of the watermarked image.

The robustness with respect to overwatermarking was tested on the watermarked 128×128 image. The image was repeatedly watermarked with up to five watermarks. The correlation values for all five overwatermarked images are shown in Figure 17. The robustness experiments together with the test for random correlations (see Figure 15) between watermarks suggest that a threshold of 0.4 should be used with the scheme.

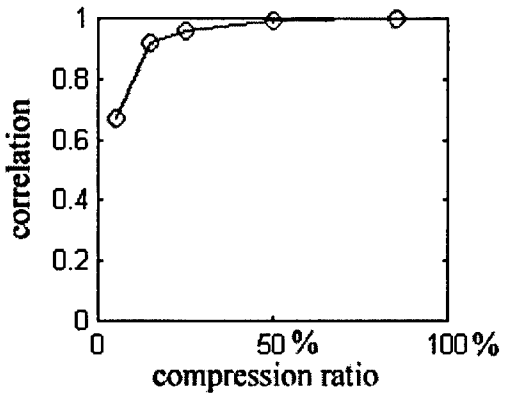


Figure 16 Robustness of global Method III to JPEG compression.

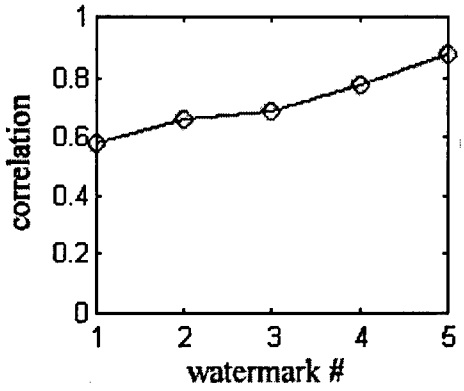


Figure 17 Robustness of global Method III to overwatermarking.

Test 2: Local scheme

In the local scheme, the image is divided into square subregions and each subregion is watermarked with a different set of orthogonal patterns. In our simulations, we used a 256×256 image of Lenna (see Figure 18) and divided it into 16 subregions of 64×64 pixels. The coefficients $\alpha(k)$ were chosen as $0.05\ (-1)^k$. The watermark length was fixed at $J = 30$ to cut down on computing time. First, the original image was watermarked (Figure 19) and then tested for presence of 100 randomly generated watermarks.

Figure 18 Original image "Lenna".

Figure 19 Watermarked image.

The robustness with respect to JPEG compression is shown in Figure 20. Comparing the correlation values to random correlations in Figure 21, it appears that the threshold of 0.25 is appropriate in this case. Using this threshold, no false detections are produced for 100 random watermarks. The threshold enables a reliable detection of 10% quality (0.38bbp) compressed JPEGs. Other image processing operations, such as blurring, noise adding, and downsampling and their combinations have been studied. The results are shown in Table 5. To test the robustness with respect to the collusion attack, the total of six images watermarked by different marks were averaged. The correlation coefficients were in the range from 0.51 to 0.71.

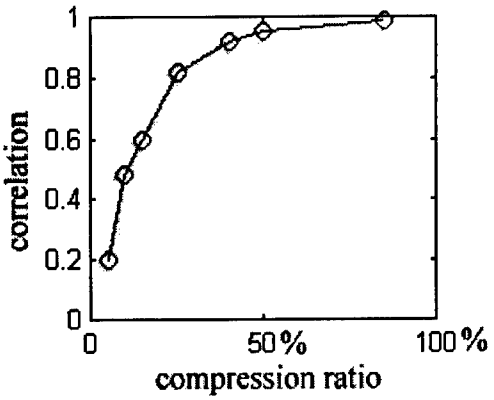


Figure 20 Robustness of local Method III to JPEG compression.

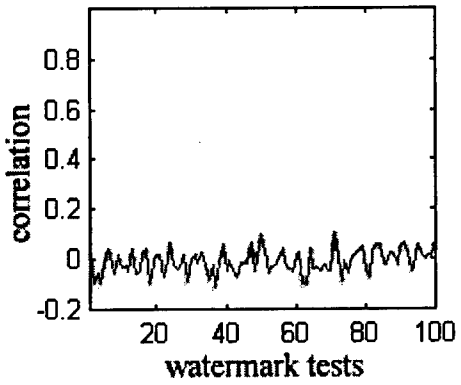


Figure 21 Correlation for 100 random watermarks for global Method III.

Table 5 Robustness of local Method III with respect to image processing operations.

Image operation	Correlation
Blurring (in PaintShop Pro 4.12)	0.68
16% uniform noise (as in PSP 4.12)	0.76
Downsampling by a factor of 2	0.53

2.2.4 Summary - comparison of watermarking methods I - III.

It would be very useful if there was a standard algorithm for comparing different watermarking methods in an objective manner. Clearly, one should only compare techniques that address the same problem and serve the same purpose. It would not be fair, for example, to compare watermarking techniques that need the original image for watermark extraction with those that do not. However, even if we constrain ourselves to techniques from one category, different watermarking techniques are based on different principles and usually

depend on several parameters that can be adjusted. The parameters determine the strength and visibility of the watermark, and the watermark size. In order to compare two techniques, one would have to adjust the watermark strength to the Just Noticeable Difference[6] (JND) in the original image and apply a standard set of image deformations. This should be repeated with a large database of various images and average the results over the images. Each standard image deformation should depend on one scalar value that can be adjusted. This will enable us to quantify the image deformation using one number and register at which value the watermark becomes undetectable. Examples of such scalar quantities are compression ratio for lossy compression scheme, the standard deviation of a Gaussian blur kernel, signal-to-noise ratio, etc. Since the perceptual measures needed for calculating the JND should be available at the end of Phase II, we plan to return to the problem of comparing watermarking techniques later during our Phase II effort.

2.3 A New Attack on Watermarking Schemes

We will describe a possible weakness of the watermarking method developed by a group at NEC. In particular, we proposed an attack based on calculating the modifications to the lowest frequency DC coefficients utilizing image areas with almost constant luminance or luminance gradient. In this report, we summarize the results of numerical experiments performed on a real image. We took a 128×128 gray scale image with 256 shades of gray. The image was modified so that a small area of pixels in the upper right corner has constant luminance of 192 (see Figure 22).

Figure 22 A test image with a small area of pixels of constant brightness (the upper right corner)

The implementation of the NEC watermarking method was realized as an m-function in Matlab. The Matlab code is included below.

```
function [Mwat,ind]=mark(M,alpha,R)
```

[6]The perceptual measures to be developed during Phase II could be used to adjust the parameters so that watermarks of approximately the same visibility are embedded by each technique to be compared.

```
[aux,N]=size(R);
t(1)=1;
i=1;

while t(i) < N,
  i=i+1;
  t(i)=t(i-1)+i;
end

count=1;
j=2;
ind(1,1)=1;
ind(1,2)=1;

while count<N & j<=i
  k=0;
  while count<N & k<=j-1
    count=count+1;
    if mod(j,2)==0
      ind(t(j-1)+1+k,1)=k+1;
      ind(t(j-1)+1+k,2)=j-k;
    else
      ind(t(j-1)+1+k,1)=j-k;
      ind(t(j-1)+1+k,2)=k+1;
    end
    k=k+1;
  end
  j=j+1;
end

XM=dct2(M);
XMwat=XM;

for i=2:N-1
```

```
        XMwat(ind(i,1),ind(i,2))=XM(ind(i,1),ind(i,2))*(1+al
        pha*R(1,i));
    end


    Mwat=idct2(XMwat);
```

The function `mark` has three input parameters and two output parameters. The input parameters are:

(1) the image `M` stored as an intensity matrix,

(2) the visibility / strength parameter `alpha`, and

(3) a vector `R` of `N` Gaussian random numbers.

The output variables denote the watermarked image `Mwat` stored as an intensity matrix, and the indexes `ind(1:N,1:2)` of modified DC coefficients (the k-th coefficient corresponds to the element `(ind(k,1),ind(k,2))` of the DCT matrix). The function first calculates the indexes, which are determined by scanning the DCT by diagonals as in the JPEG compression algorithm. Next, a DCT of the image `M` is calculated, and the lowest `N` coefficients of the DCT determined by `ind(1:N,1:2)` are modified by `alpha*R` percent, where `R` is an `N` dimensional vector consisting of `N` Gaussian random numbers with unit variance and zero mean. Finally, the inverse DCT is performed and the watermarked image `Mwat` is obtained.

The detection of the watermark is based on solving a system of P equations for `N` unknowns, where P is the number of pixels with known values of gray levels (or at least values that can be guessed based on interpretation of the image) and `N` is the length of the Gaussian random sequence (the watermark). Writing the inverse cosine transformation as

$$\mathrm{M}(i,j)=\frac{2}{\sqrt{128\times 128}}\sum_{r=1}^{128}\sum_{s=1}^{128} w_1(r)w_2(s)D(r,s)\cos\frac{\pi}{2\times 128}r(2i+1)\cos\frac{\pi}{2\times 128}s(2j+1),$$

where

$$w_1(r)=1/\sqrt{2} \text{ when } r=0 \text{ and } w_1(r)=1 \text{ otherwise}$$
$$w_2(s)=1/\sqrt{2} \text{ when } s=0 \text{ and } w_2(s)=1 \text{ otherwise}$$

and $D(r,s)$ denotes the DCT matrix of coefficients. By taking the difference between the watermarked and unwatermarked `Mwat–M`, all terms in the summation cancel except those that have been modified by the watermarking technique. Therefore, we can write for the difference `Mwat–M`

$$(\mathtt{Mwat-M})(i,j) =$$

$$\frac{2}{\sqrt{128\times128}}\sum_{k=1}^{N} w_1(r_k)w_2(s_k)\cdot\mathtt{alpha}\cdot\mathtt{R}(k)\cdot D(r_k,s_k)\cos\frac{\pi}{2\times128}r_k(2i+1)\cos\frac{\pi}{2\times128}s_k(2j+1)$$

where r_k and s_k are determined from the indexes `ind(1:N,1:2)` as r_k = `ind(k,1)` and s_k = `ind(k,2)`. The last equation is a linear system for N unknowns R(k). The number of equations is the same as the number, P, of pixels (i, j) for which the original gray level can be estimated or is known. Even though the number of pixels, P, may exceed N, the system cannot be readily solved for the watermark because the discrete cosines do not generally form a set of linearly independent functions on proper subsets of the image. In our experiment, we took P=862 pixels that had constant brightness in the original unwatermarked image. Then, we inserted a watermark into the lowest 50 DC coefficients using the function `mark`. The resulting overdetermined system of equations was solved for R(k). The original and recovered watermark sequence is shown in Figure 23. It can be clearly seen that the watermark has been identified almost exactly! It was not identified completely accurately because the matrix of the system of equations is usually ill conditioned. The Matlab's left matrix division was used to calculate the watermark sequence.

By increasing the number of modified coefficients in watermark embedding, this attack becomes harder to perform because the rank of the matrix is basically determined by the number of pixels, P, and their spatial arrangement. By increasing the number of modified coefficients, N, to 100, the Matlab linear solver could not recover the watermark sequence due to a very ill-conditioned matrix. It is not surprising that a general linear system solver breaks down in such cases. More sophisticated techniques that were not investigated so far by the PI could be put to work. For example, one could add other constraints that could make the problem of finding the watermark sequence better conditioned. One obvious possibility is to use a stabilizing functional[7] that would give penalty to sequences that do not satisfy Gaussian statistics. Even though such methods are usually computationally expensive, a speed is obviously not a critical issue in watermark breaking.

2.3.1 Analysis of the StirMark attack

One of the most powerful general watermark removing software routines for testing robustness of watermarks is the StirMark [19]. This attack is not targeted to a specific watermarking method and can be mounted against any watermarking scheme. It attempts to simulate distortions caused by scanners and copying machines. Minor geometric distortions are applied to the image and the image is resampled. In addition, a small, smoothly distributed error is added to all pixels. One can say that StirMark is a combination of a general, nonlinear geometric transformation combined with a smooth noise. StirMark has been reported to be very effective against many watermarking techniques especially when applied iteratively [19].

[7]Such methods have been well studied in image reconstruction.

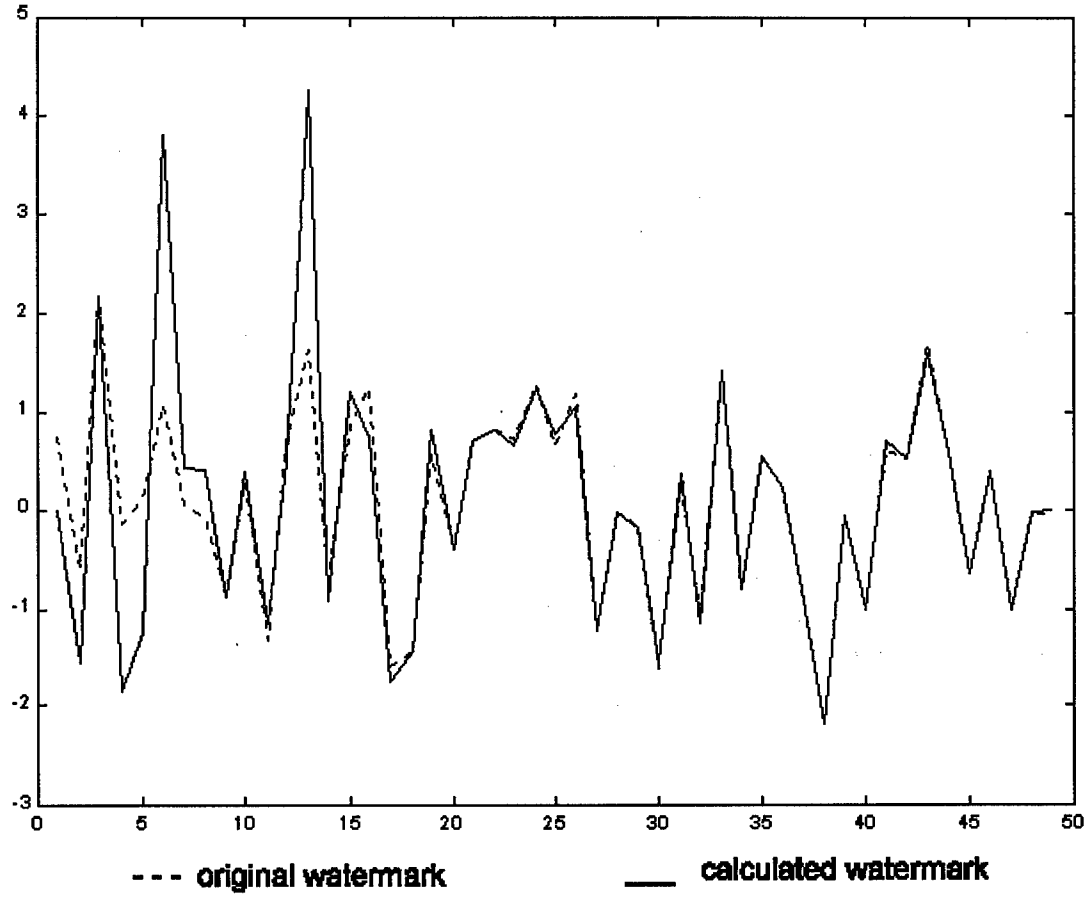


Figure 23. Comparison of the original (broken line) and the recovered (solid line) watermark of length 50.

We analyzed the effects caused by StirMark based on a DCT of the image. Previous attacks on watermarking schemes concentrated on manipulating the gray levels of the watermarked image. StirMark modifies the gray levels very little but applies a geometrical distortion to the image. As if the image was painted on a rubber canvas, it is stretched and resampled to the same pixel dimensions as the original watermarked image. This type of "horizontal" distortion, as opposed to a "vertical" distortion based on manipulating just the gray values, introduces large differences in the coefficients of the DCT. Thus, all methods that modify the coefficients of DCT of the whole image will be successfully attacked. Localized methods that embed a large number of smaller marks into the image stand a better chance of surviving the attack, but encounter serious synchronization problems. One needs to synchronize the watermark detector for each block before a cross correlation can be calculated.

The watermarking methods that are least affected are those that require the original image for watermark detection. If the original or the watermarked image is available, one can consider the nonlinear geometric distortion introduced by StirMark as a generalized type of geometric distortions, such as rotation or cropping. It then should be possible to derive the field of motion vectors and transform the tampered image to eliminate the distortion.

Figure 24 shows the influence of StirMark on the DCT coefficients. On the x axis is the relative change of DCT coefficients, while the y axis shows the percentage of the lowest 1000 coefficients that have undergone a change of at least x. For example, three applications of StirMark change 80% of the lowest 1000 DCT coefficients by at least 10%, while roughly 30% of coefficients have been modified by at least 50%, etc. The graph compares the modifications with other image processing operations, such as JPEG compression, blurring, downsampling, and noise adding. Extreme modifications of the image, such as multiple blurring, 5% quality JPEG compression (0.38bpp), or 4x downsampling (this eliminates almost 94% of all information from the image!), cause extremely visible degradations but they do not modify DCT coefficients as severely as one application of StirMark! This means that StirMark is able to modify the Fourier spectrum of the image quite extensively while keeping the perceptual quality of the image at a very reasonable level. This is caused by the nonlinear distortion of the image. For comparison, in Figure 25 we plotted the original image and images after one, two, and three applications of StirMark. A casual inspection does not reveal any big differences. Under a closer inspection, one can clearly see that the image has been stretched along the horizontal direction. Lenna's hat has been visibly stretched, and the distortion becomes very visible at the edges. Thus, StirMark achieves a perceptually non-disturbing distortion while making relatively big changes to the Fourier spectrum. This is caused by the nonlinear geometrical transformation.

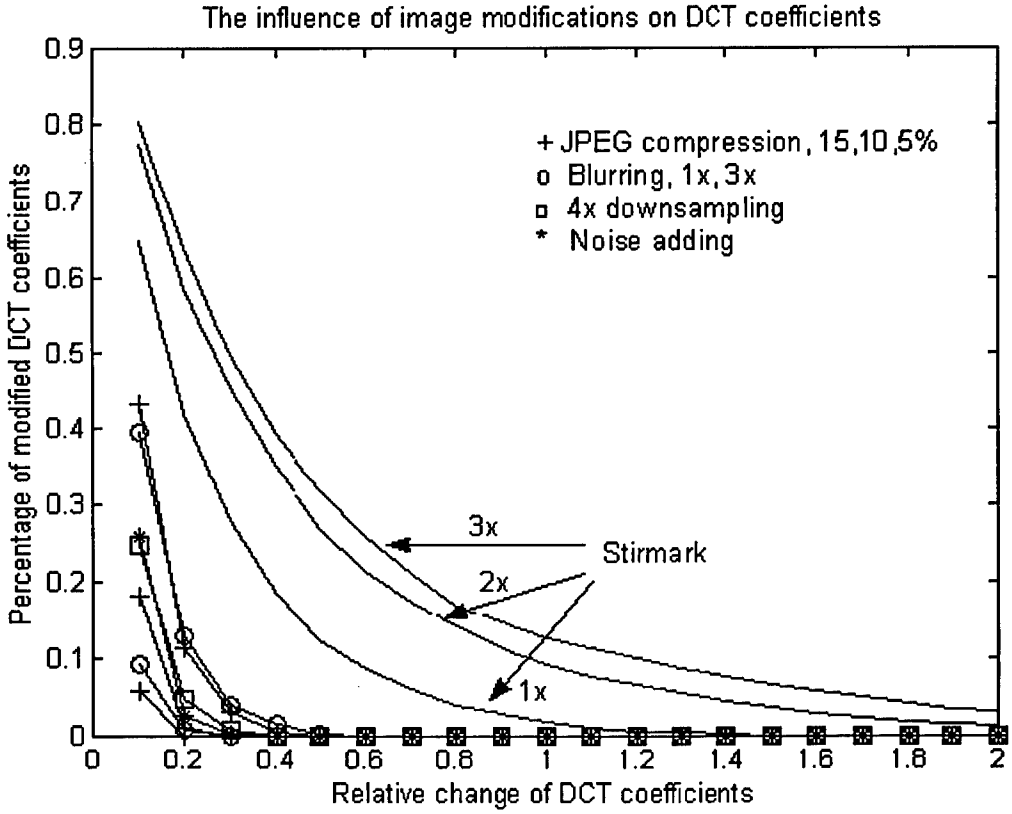


Figure 24 The influence of StirMark on DCT coefficients.

Original image

Stirmark 1x

Stirmark 2x

Stirmark 3x

Figure 25 Results of several applications of StirMark.

2.3.2 A countermeasure for StirMark attack

As argued above, if the original image is available or if at least the untampered watermarked image is available, we may try to remap the StirMark tampered image back before we apply any watermark detector. This is, in fact, necessary. It is a commonly accepted fact that geometrical transformations, such as rotation, cropping, or affine transformation should be applied before any watermark detector can be applied. There is no reason, why the same logic should not be applied for a more general type of geometric distortion introduced by StirMark.

We have implemented a simple *C* routine, *adjust.c*, that finds motion vectors between two images and adjusts one image according to the other image.

The routine needs four line parameters as the input: the original image *im*1, the tampered image *im*2, a positive integer *r*, and a second positive integer *S*. The images must be in the Portable Gray Map (PGM) format. This raw image format is easy to manipulate, read, and write. The integer *r* specifies the size of a square region that will be matched between the images *im*1 and *im*2. The second integer *S* specifies the size of the square region in which the program searches for a matching region. The pseudo code for *adjust.c* is shown below. The routine produces an adjusted image *adj.pgm*.

The program never uses the gray levels of the original image. Only the gray levels of the tampered image are used for the new, adjusted image. The original is only used to get the spatial correlation and invert the geometric distortion.

The routine described above was used to test the robustness of the watermarking technique based on orthogonal patterns developed in our effort from the previous month.

```
begin
x = im1; w = im2;  // Read the images, convert them to matrices with entries in [0,1]
for i = r+S to N−r−S
  for j = r+S to M−r−S
    for u = −S to S
      for v = −S to S
        match the squares x[i−r...i+r][j−r...j+r] and w[u−r...u+r][v−r...v+r]
        // The size of both squares is (2r+1) × (2r+1)
        // Matching could be based on the difference between the two squares or
        // the correlation between them
      end (v)
    end (u)
    find the best matching square (U,V) out of those (2S+1)^2 squares
    set adj[i][j] = w[U][V]      // adj[][] is the adjusted image
  end (j)
end (i)
```

When StirMark was applied only once, the value of the correlation was barely above the detection threshold set to 0.2 (see Table 6). Two and three applications of StirMark without adjustment successfully removed the watermark (Table 6). With the adjustment, the values of correlation were well above threshold.

We also combined the StirMark attack with other image processing operations such as lossy JPEG compression with quality factors as low as 15%, blurring, and noise adding. The results are shown in Table 6.

Watermarking techniques that do not require the original image will encounter synchronization problems when applied to images modified with StirMark. It appears that localized techniques with small block sizes will help overcome the synchronization problem at the expense of computational complexity.

Table 6 Watermark detection with and without adjustment for StirMark.

Distortion	Correlation without adjustment	Correlation with adjustment
StirMark 1x	0.26	0.78
StirMark 2x	0.14	0.81
StirMark 3x	0.10	0.78
StirMark 3x + 20% uniform noise (as in PSP 4.12)	< 0.1	0.53
StirMark 3x + blurring 1x in PSP 4.12	< 0.1	0.69
StirMark 3x + JPEG 25% quality compression	< 0.1	0.79
StirMark 3x + JPEG 15% quality compression	< 0.1	0.78

2.4 Implementation of Watermarking Techniques

2.4.1 Basics of Matlab

Run Matlab by double clicking the Matlab icon in Win 95 or NT. Change the default-working directory from c:\Matlab\bin to some working directory of your choice. You can choose either DOS or UNIX commands, such as *dir, cd, ls, pwd*, etc. Copy the images that you plan to watermark and the software provided on the enclosed floppy to your working directory. Make sure that you have installed the image processing toolbox before you start with running watermarking applications. The commands for importing, viewing, and exporting images are:

[X,map]=bmpread('image.bmp');	for importing image from a disk
imshow(X,map);	for viewing an indexed image in Matlab
bmpwrite(X,map,'image.bmp');	for writing an image to disk

More detailed information can be found in Matlab's manual or using the line command *help*.

2.4.2 Watermarking Method I

In Section 2.2.1, we described and tested a watermarking technique based on pattern overlaying. The original image is watermarked with a pattern that is generated in several steps. First, the hash of the original image together with userID number and other appropriate information is used to seed a PRNG to generate an initial black and white pattern. The pattern is further processed using a cellular automaton and smoothened to push the watermark's

energy into low frequencies. The final pattern is rescaled and added to the original image. To test the presence of a specific watermark in a modified image, the original image is subtracted and a cross-correlation in the Fourier space is used to decide about the presence of the watermark.

The implementation in Matlab has been designed for gray scale images with arbitrary dimensions. The extension to color images is straightforward – one could watermark each channel with a different watermark, or switch to a more useful color base, such as Luminance, Hue, and Saturation, and watermark the luminance, converting the image back to RGB after the watermarking is completed. We further assume that all the images involved are in the Windows BMP format. It is possible to work with more general formats, such as TIFF, but we opted for the simple BMP format because the main purpose of this implementation is proof of concept and to provide routines for testing the methods.

Let us assume that we have an original image *orig.bmp* stored in the working directory. Make sure you have the following m-functions in the same working directory: *watermark.m* and *detect.m.* To watermark *orig.bmp* with a user-defined watermark, simply use the following line command:

[Y,w]=watermark('orig.bmp', seed);

where *Y* is the watermarked image matrix, *w* is the watermark matrix, and *seed* is the seed for Matlab's PRNG generator. In the final implementation, the seed will be automatically determined from the original image, author's ID, and other information. The matrices *Y* and *w* are outputs of the watermark m-function and will be automatically generated. The watermarked image *Y* is stored as *water.bmp* in the working directory. The stages of the watermarking process are displayed as the calculations proceed. The function renders both the original image and the watermarked image on the computer screen automatically (two Matlab Figures automatically pop up). Although the speed of the algorithm is very reasonable, we expect C/C++ implementations to run even faster.

To test the presence of the watermark *w* in image *test.bmp*, type the following at the Matlab prompt:

detect('test.bmp', 'orig.bmp', w)

without a ";" at the end. The result will be the correlation number between 0 and 1. Based on the tests described in Section 2.2.1, we set the threshold for watermark detection to 0.3. This number is based on the highest correlation value (0.17) obtained for detection of random watermarks in an unwatermarked image.

At the end of Section 2.2.2, we argued that the reliability of watermarking method II needs to be improved before a working prototype software can be offered for further testing and demonstrations. This is why the Matlab code for the watermarking method II is not provided at this point.

2.4.3 Watermarking Method III

The theory behind watermarking based on random smooth orthogonal patterns has been presented in our monthly report No. 7 and 8. The practical implementation in Matlab is very

similar to the previous method. Before you start, make sure you have the following m-functions in your working directory:

watermark.m, detect.m, genog.m, mark.m, gs.m, and *smooth.m.*

Note: The *watermark* and *detect* functions are not the same functions as in the watermarking method II. We recommend that you create a new working directory for each watermarking technique to avoid using incorrect m-functions.

To watermark the original image *orig.bmp*, type

Y=watermark('orig.bmp', Blocksize, Patterns, seed);

where *Blocksize* is the size of the blocks used for embedding watermarks (see Section 2.2.3), and *Patterns* is the number of orthogonal patterns. The choice of these parameters is important and will determine the robustness and processing speed. For a 256×256 image, we recommend *Blocksize*=64, *Patterns*=30. For a general image, we recommend to choose the *Blocksize* to be at most one quarter of the image size, but probably not smaller than 16. Keep the number of patterns at 30 or higher. The watermarked image is stored as *water.bmp* in your working directory. The function *watermark* renders both the original image and the watermarked image on the computer screen automatically (two Matlab Figures automatically pop up).

To detect a watermark, use this command:

detect('orig.bmp', 'water.bmp', 'test.bmp', Blocksize, Patterns, seed)

again without the semicolon at the end. The result is the value of correlation of the watermark in the image *test.bmp*. Note that unlike in the previous method, the watermark is not stored in the variable *w*. Actually, it is not explicitly stored in any variable. This is because we would have to store all orthogonal patterns and that would take too much space. It is actually more convenient (and secure!) to let the detect function compute the orthogonal patterns for each detection. The inner workings of all the Matlab routines are explained using comments inside the bodies of corresponding m-functions.

3. Recommendations for Future Research

There are several research direction one can pursue to further extend techniques developed in Phase I. The research directions detailed below have already been proposed for Phase II research.

The steganographic techniques described in Section 2.1 should be further extended so that the capacity of the hidden channel can be maximized while keeping a specified level of security. The security level could be measured using a set of statistical measures. The measures can be used to quantify the modifications of the carrier image caused by hiding the secret message. This way, one will be able to choose the best carrier image to fit a given message. Also, by changing the requirements on the statistical measures that should be preserved by the hiding algorithm, one can opt for an extremely secure method while sacrificing the capacity (for messages of extreme importance), or choose a large capacity at the expense of security (for less security-demanding applications).

Another major research direction that will be pursued in the future is the extension of robust message hiding algorithms to other bases than Fourier trigonometric bases. By making the choice of the basis key-dependent, the security of the technique significantly increases while the capacity of the hidden channel also grows. One possibility would be to use wavelet bases as a replacement for DCT. We expect that the flexibility of wavelets will generally provide higher embedding capacity.

Utilization of the properties of the HVS seems necessary for capacity-optimized data embedding techniques. The irrelevant (invisible) subspace of typical images appears to be large enough to allow multiple bit embedding for subregions as small as 8×8 pixels. It will be interesting to see if current low-bit encoding schemes that utilize the properties of the HVS pose a threat to such watermarking schemes. An important tool for successful development of watermarking schemes based on the properties of the HVS, is a good model of the HVS, or, equivalently, a good perceptual measure between images.

To quantify the robustness of message hiding schemes, a standardized set of image processing tools should be employed. It should be possible to quantify the security with a set of global and local statistical measures, enabling thus objective comparison of security of different message hiding schemes and their evaluation.

Future research on robust data embedding will bring solutions to an important and difficult problem of image access control. If a standard for marking image content were accepted, the construction of "intelligent" Internet browsers capable of filtering out images with certain marks would become possible. This way, certain images will not be displayed depending on user's login name. For example, one could filter out images inappropriate for viewing by children or images that should be available only to people with certain security clearance.

4. Applications

The techniques developed and investigated during Phase I will be further extended and developed in Phase II. The anticipated research results of Phase II are:

- Techniques for robust hiding of messages in digital imagery based on wavelet bases and general incomplete orthogonal bases;
- Perceptual measures for determining the visibility of the watermark inside an image;
- Set of global and local statistical measures and integral measures for testing the security of message hiding schemes;
- Adaptive message hiding schemes for hiding large messages inside digital images while preserving a specified level of security (a set of statistical measures);
- A prototype software product using which a user can (a) watermark his image in an extremely robust way to protect copyright, fingerprint the image, or communicate a short message in a robust way, or (b) hide a large message within a digital image with a certain security level.

The anticipated tasks for Phase III include the following activities:

(1) Marketing the patents developed during the research on Phase I and Phase II;

(2) Marketing the prototype software product;

(3) Writing plug-ins for current wide spread image manipulating software packages such as PhotoShop, PaintShop, Corel Draw, etc.;

(4) Creating an on-line commercial watermarking web-service for image marking;

(5) Creating a commercial software package for image marking and covert communication, which will include message compression, encryption and hiding.

It is to be expected that the importance of secret message hiding in digital imagery will only increase due to the following factors:

- Rapidly developing computer networks, such as the Internet;
- Ever increasing need to transmit images, graphs, charts, sound, and video;
- Emergence of new technologies such as videophone, HDTV, digital surveillance video-cameras;
- Increasing bandwidth of communication links;
- Growing number of subjects communicating via computer networks;
- Availability of images from satellites, airplanes, soldier-carried cameras;
- Use of digital imagery in the court of law.

The techniques will be directly applicable to secure intelligence data manipulation, information protection, and information warfare. Besides obvious military applications, the private sector will substantially benefit from the proposed techniques. For example, watermarks can be used for solving a dispute about an ownership of some digital data, for fraud protection, distribution of digital video, and increasing the bandwidth of existing communication links (adding subtitles in several languages, captions, comments, small icon with sign language, etc within the image). Other applications of the proposed techniques include:

Military Applications: Covert communication, low-probability-of-intercept communications, traitor tracing, authentication, increasing communication bandwidth (adding comments, captions, image status information), control over access to images with different degree of clearance

Commercial Applications: Copyright protection, fingerprinting, tracing of illegal copies, intelligent Internet browsers capable of automatic filtering of images with certain content, image integrity protection

Covert communication (military)

When two parties A and B are exchanging encrypted messages, they raise suspicion and attract attention of a third party who can try to break the code, attempt to find out the passwords and keys, disturb the communication channel (even infrequent and "small" errors will make decryption impossible due to cipher-text sensitivity), and cut the communication channel (refuse services).

Traitor tracing (military).

Central agency releases information to deputies A, B, C, D, E. One of them leaks the information to the enemy. By marking each of the five documents distributed to A–E with a different invisible mark, one could trace down the traitor provided the compromised information becomes available.

Authentication (military and commercial)

Classical authentication and signature schemes work by appending a message digest to the message. Given a secret message M, an encryption function of the sender f_S , and an encryption function of the receiver f_R , the sender appends $f_S^{-1}(H(M))$ (encrypted hash of M) and sends f_R (M * $f_S^{-1}(H(M))$). The receiver decrypts the message by applying f_R^{-1} and verifies that the message came from S by comparing $H(M)$ and f_S of the encrypted message digest. One potential disadvantage of this authentication scheme is that once the message has been decrypted, the signature can be removed and replaced with a different signature. In other words, the authentication scheme is for *exchanging* messages. It is part of the communication protocol. Once the message is received and decrypted, the authentication also "expires." Authentication based on image watermarking works in an entirely different way. The signature stays in the image as long as the image is recognizable, it is present even in a small part of the image (it is possible to identify the author given a portion of the image), the signature (mark) cannot be removed or replaced without a correct password (key). Thus, the watermarking scheme is much more flexible and offers many other commercial applications, such as digital watermark, fingerprinting, copyright protection of images, etc.

Intelligent Internet browsers (military and commercial).

Such browsers will have the capability of filtering out images with certain marks depending on who is logged on to the system. Images will contain marks, which could be automatically detected by the browser before displaying them on the monitor. In this manner, certain images will be available only to people with a specified degree of clearance. Similarly, adult images not appropriate for viewing by children will not be displayed to a user with a certain user ID, etc.

Captions (military and commercial).

Adding captions to images in a robust way is another important potential application of the proposed research. Caption may include but is not limited to the following information: the author of the image, place, time, sender, image content (no hash functions but, say, coefficients of the largest DCT coefficients).

Classical methods for adding captions:

Appending the caption to the image by inserting several rows of pixels. Can be easily removed or replaced, increases the size of the image. Overlaying the caption in a visible form into the image (for example into the lower right corner). The caption may disturb an important part of the image (if done automatically, otherwise requires a human to do it), can be easily removed or replaced.

New methods for caption adding using data hiding:

Embedding the caption into the image rather than appending or overlaying it offers the following advantages: The image size stays the same, the caption is invisible and does not deteriorate the image, the caption cannot be removed without a password, it stays in the image as long as the image content is not drastically changed (after lossy compression, filtering, etc.), the caption cannot be replaced or removed without the password, the caption embedding can be automated and does not require human intervention.

Proof of integrity of digital imagery for law enforcement agencies (commercial)

Creating a system for watermarking video-streams for proof of image integrity for the chain of custody for law enforcement agencies. The purpose of the watermark would be to proof that the image has not been tampered with. Therefore, the watermarked video could be used as evidence in the court of law.

Appendix A: Discrete Cosine Transformation

$$D(i,j) = \frac{2}{\sqrt{M \times N}} \sum_{r=1}^{M} \sum_{s=1}^{N} w_1(r) w_2(s) I(r,s) \cos\frac{\pi}{2 \times M} r(2i+1) \cos\frac{\pi}{2 \times N} s(2j+1),$$

where

$w_1(r) = 1/\sqrt{2}$ when $r = 0$ and $w_1(r) = 1$ otherwise

$w_2(s) = 1/\sqrt{2}$ when $s = 0$ and $w_2(s) = 1$ otherwise

the size of the image is $M \times N$

$I(r,s)$ denotes the image matrix of gray values

and $D(r,s)$ denotes the DCT matrix of coefficients.

Inverse Discrete Cosine Transformation

$$I(i,j) = \frac{2}{\sqrt{M \times N}} \sum_{r=1}^{M} \sum_{s=1}^{N} w_1(r) w_2(s) D(r,s) \cos\frac{\pi}{2 \times M} r(2i+1) \cos\frac{\pi}{2 \times N} s(2j+1),$$

References

[1] Aura T., Invisible communication. In *Proc. of the HUT Seminar on Network Security '95*, Espoo, Finland, November 1995. Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology. [http://deadlock.hut.fi/ste/ste_html.html], [ftp://saturn.hut.fi/pub/aaura/ ste1195.ps]

[2] Bender W., D. Gruhl, and N. Morimoto, "Techniques for Data Hiding." In *Proc. Of SPIE*, vol. 2420, p. 40, 1990.

[3] Brassil J., S. Low, N. Maxemchuk, L. O'Goram, "Document Marking and Identification using Both Line and Word Shifting," Infocom95. [ftp://ftp.research.att.com/dist/brassil/1995/infocom95.ps.Z]

[4] Brassil J., S. Low, N. Maxemchuk, L. O'Goram, "Electronic Marking and Identification Techniques to Discourage Document Copying," Infocom94. [ftp://ftp.research.att.com/dist/brassil/1994/infocom94a.ps.Z]

[5] Brassil J., S. Low, N. Maxemchuk, L. O'Goram, "Hiding Information in Document Images," CISS95. [ftp://ftp.research.att.com/dist/brassil/ 1995/ciss95.ps.Z]

[6] Comiskey, B. O. and J.R. Smith, "Modulation and Information Hiding in Images," in: *Information Hiding, First International Workshop*, edited by Ross J. Anderson. Cambridge, U.K., May 30–June 1, 1996, Proceedings. Lecture Notes in Computer Science, Vol. 1174, Springer-Verlag, 1996 [http://sunsite.informatik.rwth-aachen.de/dblp/db/conf/ih/ih96.html]

[7] Cox, J. I., J. Kilian, T. Leighton, and T. Shamoon, Secure Spread SpectrumWatermarking for Multimedia," NEC Research Institute, *Technical Report* 95-10.

[8] Craver, S., N. Memon, B.-L. Yeo, and M. Yeung. "Can invisible watermarks resolve rightful ownerships?" *Proceedings of the IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases V*, San Jose, CA, USA, Feb. 13–14, 1997, vol. 3022, pp. 310–321.

[9] Delaigle, J.-F., C. De Vleeschouwer, B. Macq, "Digital watermarking of images," *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging Science and Technology*, 1996.

[10] Dixon R. C., *Spread Spectrum Systems with Commercial Applications*. Wiley, New York, 1994.

[11] Fridrich, J., "On Digital Watermarks," paper for the 2nd Information Hiding Workshop in Portland, Oregon, April 15–17, 1998.

[12] Fridrich, J. "Symmetric Ciphers Based on Two-Dimensional Chaotic maps," to appear in *Int. J. Bifurcation and Chaos*, **8**(6), June 1998.

[13] Girod B. and F. Hartung, "Watermarking Method and Apparatus for Compressed Digital Video", US Patent application, 1996. [http://www-nt.e-technik.uni-erlangen.de/~hartung/watermarking.html]

[14] Handel and Sanford [http://www.lanl.gov/users/u078743/embed1.htm]

[15] Maxwell T. Sandford II, Jonathan N. Bradley, and Theodore G. Handel. The data embedding method. In *Proc. of the SPIE Photonics East Conference*, Philadelphia, September 1995.

[16] Hartung F. and B. Girod, "Digital Watermarking of Raw and Compressed Video", *Proc. European EOS/SPIE Symposium on Advanced Imaging and Network Technologies*, Berlin, Germany, Oct. 1996.

[17] Kahn D., *The Codebreakers*, The Macmillan Company. New York, NY 1967.

[18] Koch E. and J Zhao, Towards Robust and Hidden Image Copyright Labeling, *Proc. of 1995 IEEE Workshop on Nonlinear Signal and Image Processing* (Neos Marmaras, Halkidiki, Greece, June 20–22, 1995) and [http://www.igd.fhg.de/www/igd-a8/pub/IEEE_Hidden.ps]

[19] Kuhn, M.G. *Stirmark.* Available *at http://www.cl.cam.ac.uk/~mgk25/stirmark/*, Security Group, Computer Lab, Cambridge University, UK (E-mail: mkuhn@acm.org), 1997.

[20] Podilchuk, C. and W. Zeng, "Digital image watermarking using visual models," Technical Memo, Multimedia Communication Lab, Lucent Technologies, Bell Labs, Sept. 1996.

[21] Sattler, M. Home Page.

[http://www.indstate.edu/msattler/sci-tech/comp/privacy/topics/steganography.html].

[22] Schneier, B. [1996] *Applied Cryptography* (Wiley, New York).

[23] Van Schyndel, R.G., A.Z. Tirkel, N.R.A Mee, C.F. Osborne, "A digital watermark," *Proceedings of the IEEE International Conference on Image Processing*, November, 1994, Austin, Texas, USA, vol. 2, pp. 86–90.

[24] Swanson, M., B. Zhu, and A. H. Tewfik, "Data Hiding for Video in Video," in *Proc. ICIP '97*, vol. II, pp. 676–679.

[25] Tewfik, A.H., M.D. Swanson, B. Zhu, K. Hamdy, and L. Boney, "Transparent Robust Watermarking for Images and Audio." *IEEE Trans. on Signal Proc.*, 1996.

[26] Bob Tinsley [http://www.sun.rhbnc.ac.uk/~phac107/]

[27] Toffoli, T. and N. Margolus, *Cellular Automata Machines*, MIT Press, 1987.

[28] *Unzign*. Available at *http://altern.org/watermark/* (E-mail: unzign@hotmail.com), 1997.

[29] Zhao, J. and E. Koch, Embedding Robust Labels Into Images For Copyright Protection, *Proc. Int. Congr. on IPR for Specialized Information, Knowledge and New Technologies* (Vienna, Austria, August 21–25, 1995) [http://www.igd.fhg.de/www/igd-a8/pub/EmbedLabel.ps]

*U.S. GOVERNMENT PRINTING OFFICE: 1998-610-130-81059

MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.